US006154213A

United States Patent [19]

Rennison et al.

[11] Patent Number: 6,154,213

[45] Date of Patent: Nov. 28, 2000

[54] **IMMERSIVE MOVEMENT-BASED INTERACTION WITH LARGE COMPLEX INFORMATION STRUCTURES**

[76] Inventors: **Earl F. Rennison**, 1076 De Haro St., San Francisco, Calif. 94107; **Lisa S. Strausfeld**, 2355 Polk St., San Francisco, Calif. 94109; **Damon M. Horowitz**, 130 Frederick St., #106, San Francisco, Calif. 94117

[21] Appl. No.: **09/087,259**

[22] Filed: **May 29, 1998**

Related U.S. Application Data

[60] Provisional application No. 60/048,150, May 30, 1997.

[51] **Int. Cl.**[7] **G06F 3/14**

[52] **U.S. Cl.** **345/356**; 345/357; 345/334; 345/349; 345/428; 345/333; 707/103; 707/104; 707/501

[58] **Field of Search** 345/356, 353, 345/357, 348, 349, 333, 334, 428; 707/103, 104, 501, 514

[56] **References Cited**

U.S. PATENT DOCUMENTS

5,008,853	4/1991	Bly et al.	345/331
5,062,060	10/1991	Kolnick	345/339
5,241,671	8/1993	Reed et al.	707/104
5,481,666	1/1996	Nguyen et al.	345/357
5,537,526	7/1996	Anderson et al.	707/515
5,544,302	8/1996	Nguyen	345/348
5,550,563	8/1996	Matheny et al.	345/348
5,557,722	9/1996	Derose et al.	345/357
5,584,035	12/1996	Duggan et al.	345/339
5,623,589	4/1997	Needham et al.	707/501
5,675,752	10/1997	Scott et al.	345/352
5,721,851	2/1998	Cline et al.	345/357
5,832,494	11/1998	Egger et al.	707/104
5,877,766	3/1999	Bates et al.	345/357
5,978,811	11/1999	Smiley	707/104

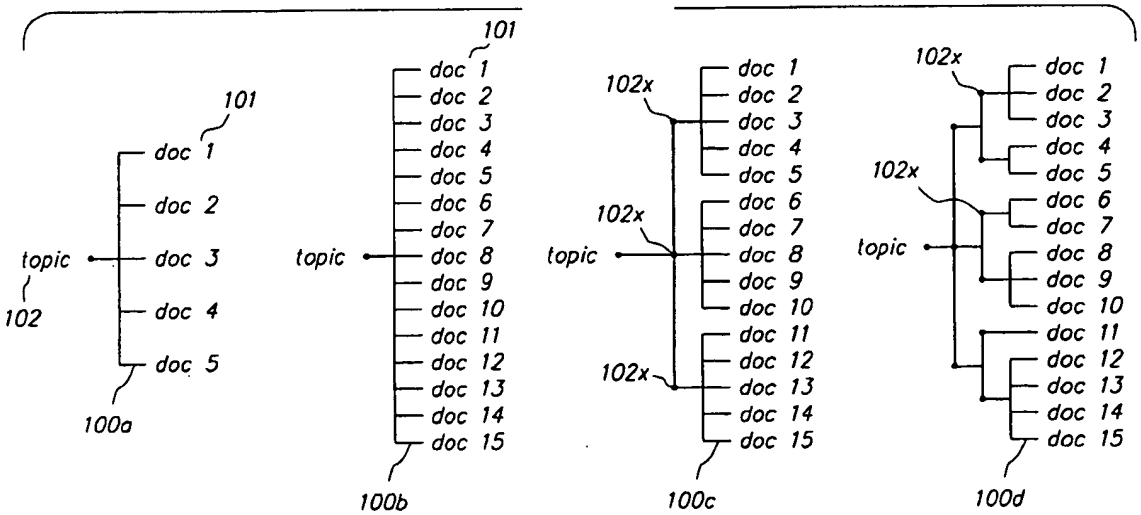
Primary Examiner—Raymond J. Bayerl
Assistant Examiner—Thomas T. Nguyen
Attorney, Agent, or Firm—Fenwick & West LLP

[57] **ABSTRACT**

An intuitive, immersive, movement-based interface and system provides for navigating through large collections of multidimensional information. The interface allows users to navigate through large document collections by maintaining a constant density of visual information presented on a display device to the user at any given moment of time. The document collection is organized in an immersive information space, containing various levels of topics and related documents. At each level within the immersive information space contextual information is presented to the user. The contextual information consists of a semantic scale and a pathway to the information they are viewing. An information structure represents the immersive information space of documents. The information structure consists of a collection of documents, and a graph of topics that describe the relationships between the documents. The graph of topics consists of topic nodes that each contain 1) a set of documents that are about that topic, and 2) a set of links to other topics in the structure. The links represent relationships between topics, and indirectly, relationships between the documents. An information structure that represents the collection of documents is used to guide the user to documents of interest and to show relationships between documents. A presentation and interaction model allows navigation through the information structure. The model includes a camera representing a user's focus of attention, and a set of reactable graphical objects representing nodes in the information structure. The interaction model continuously monitors the movement of the camera in relation to the graphical objects and updates the display of the information space.

11 Claims, 9 Drawing Sheets

Microfiche Appendix Included
(3 Microfiche, 216 Pages)



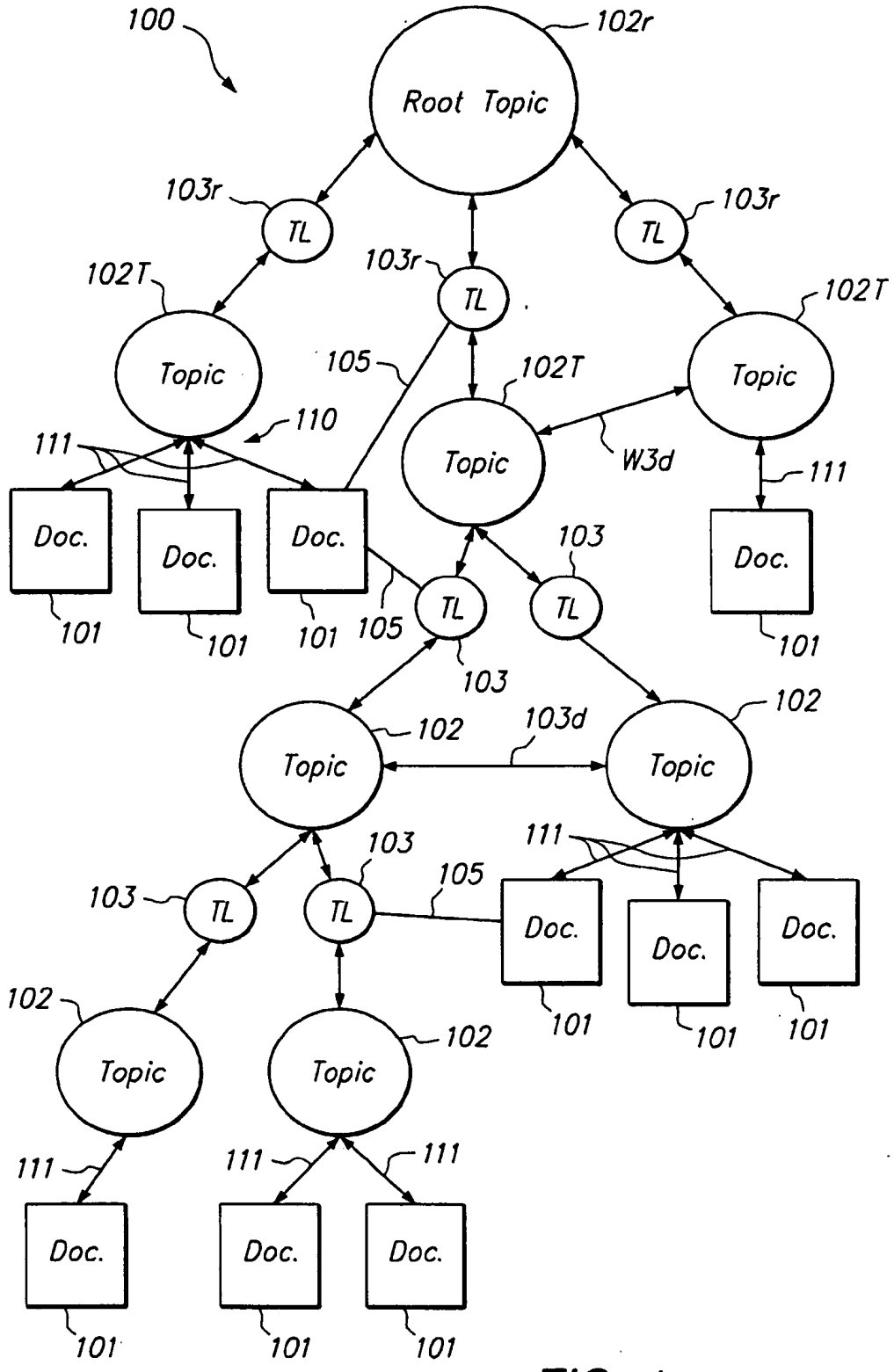

100
102r
Root Topic
103r
TL
103r
TL
103r
TL
102T
Topic
105
102T
Topic
102T
Topic
110
W3d
111
111
Doc.
Doc.
Doc.
Doc.
101
101
101
105
TL
TL
103
103
102
103d
102
Topic
Topic
103
111
103
105
TL
TL
Doc.
Doc.
Doc.
102
101
101
101
Topic
Topic
102
111
111
111
Doc.
Doc.
Doc.
101
101
101

FIG. 1

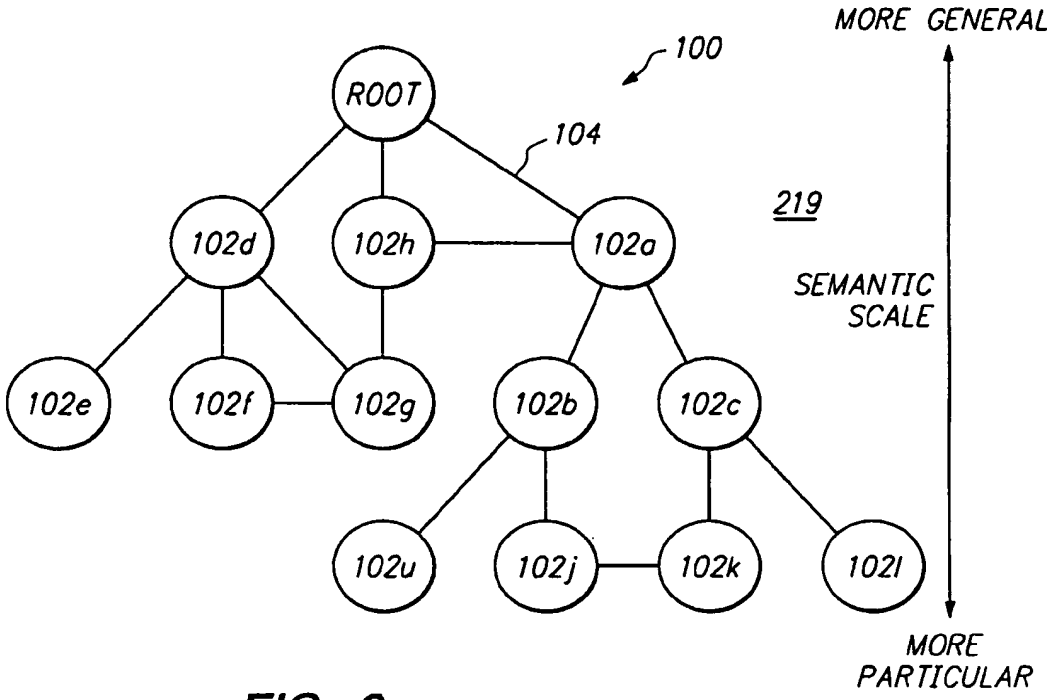

MORE GENERAL
100
ROOT
104
219
102d
102h
102a
SEMANTIC SCALE
102e
102f
102g
102b
102c
102u
102j
102k
102l
MORE PARTICULAR

FIG. 2

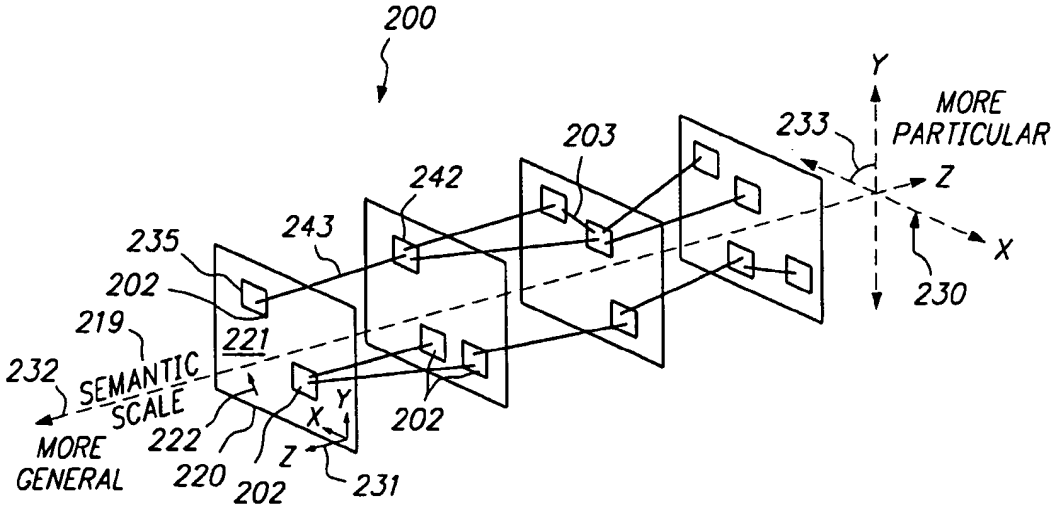

200
Y
MORE PARTICULAR
233
203
242
243
Z
235
202
219
221
X
230
232
SEMANTIC SCALE
MORE GENERAL
222
220
202
231
202

FIG. 3

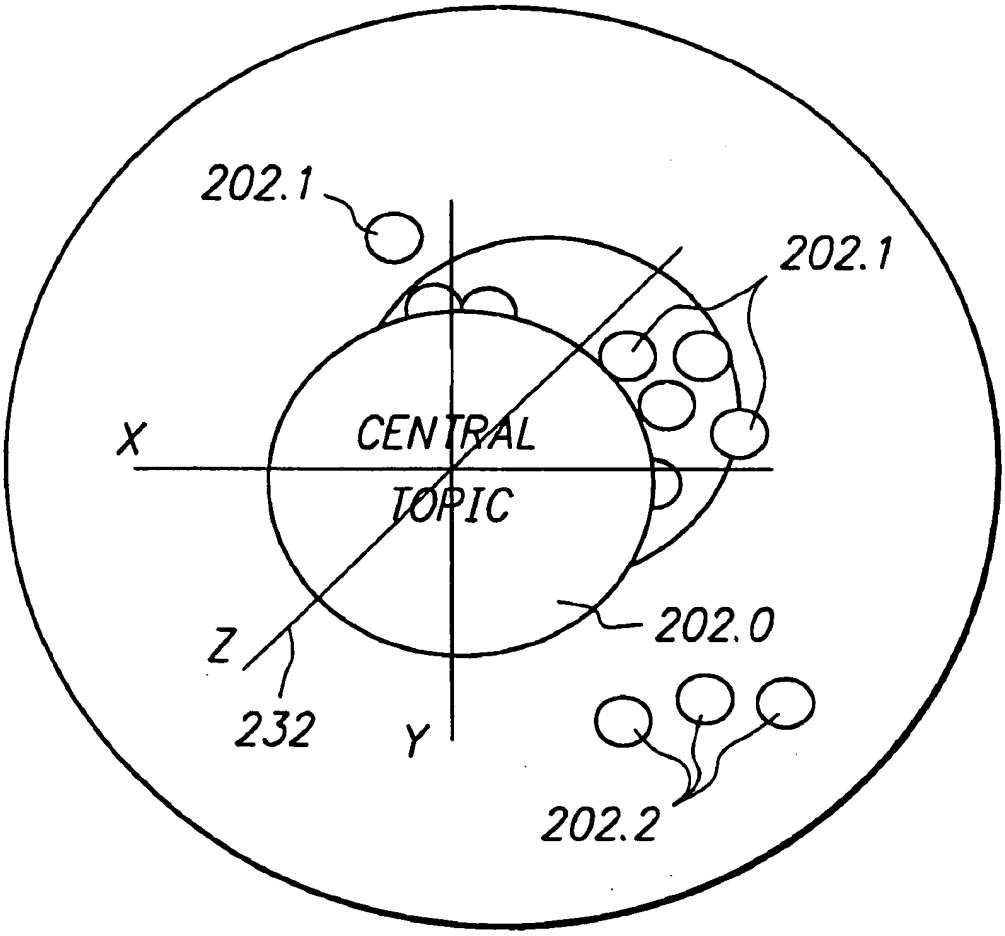
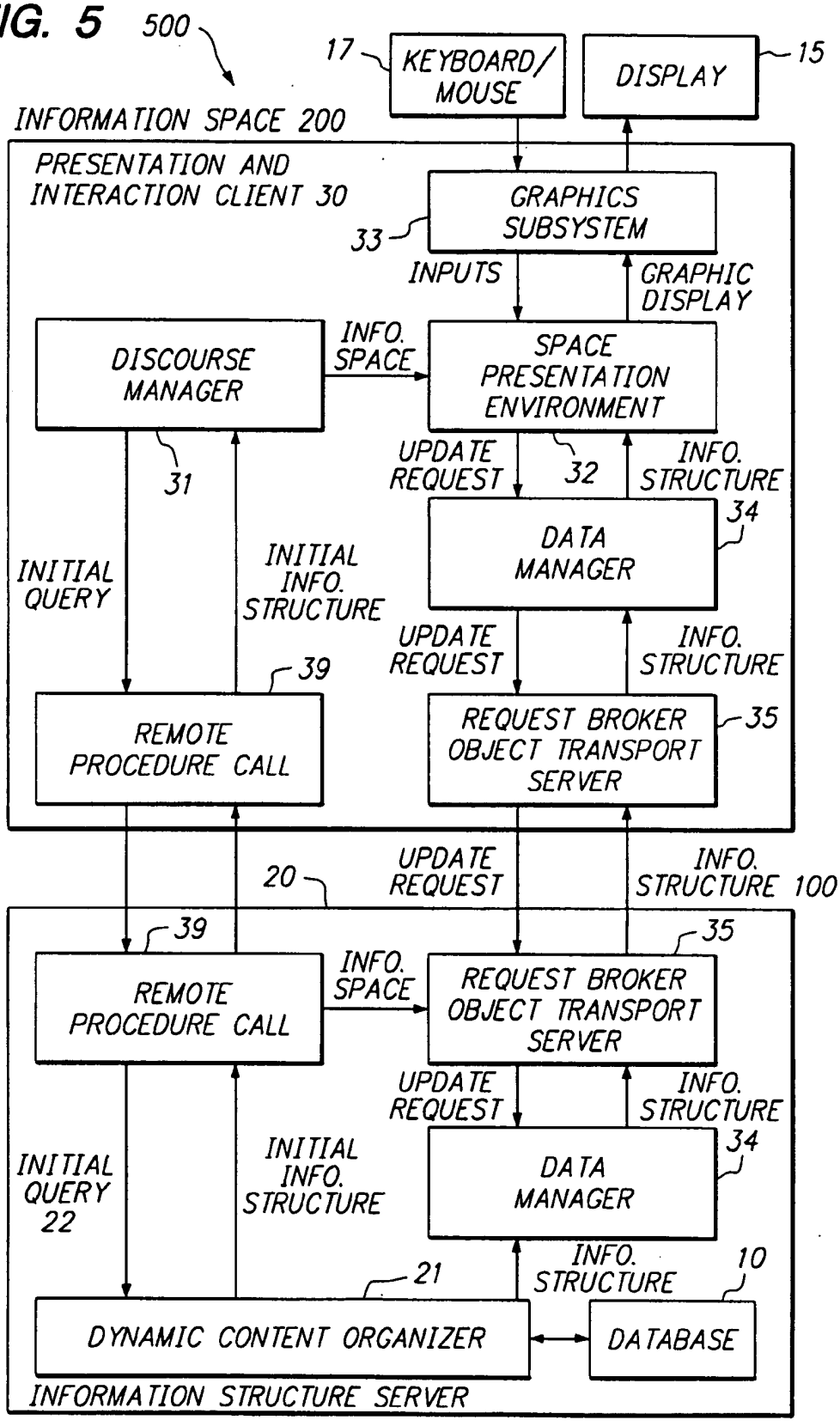


FIG. 4

FIG. 5


500
17
KEYBOARD/ MOUSE
DISPLAY
15
INFORMATION SPACE 200
PRESENTATION AND INTERACTION CLIENT 30
GRAPHICS SUBSYSTEM
33
INPUTS
GRAPHIC DISPLAY
DISCOURSE MANAGER
INFO. SPACE
SPACE PRESENTATION ENVIRONMENT
31
UPDATE REQUEST
32
INFO. STRUCTURE
34
DATA MANAGER
INITIAL QUERY
INITIAL INFO. STRUCTURE
UPDATE REQUEST
INFO. STRUCTURE
39
REMOTE PROCEDURE CALL
REQUEST BROKER OBJECT TRANSPORT SERVER
35
20
UPDATE REQUEST
INFO. STRUCTURE 100
39
35
REMOTE PROCEDURE CALL
INFO. SPACE
REQUEST BROKER OBJECT TRANSPORT SERVER
UPDATE REQUEST
INFO. STRUCTURE
INITIAL QUERY 22
INITIAL INFO. STRUCTURE
34
DATA MANAGER
INFO. STRUCTURE
10
21
DYNAMIC CONTENT ORGANIZER
DATABASE
INFORMATION STRUCTURE SERVER

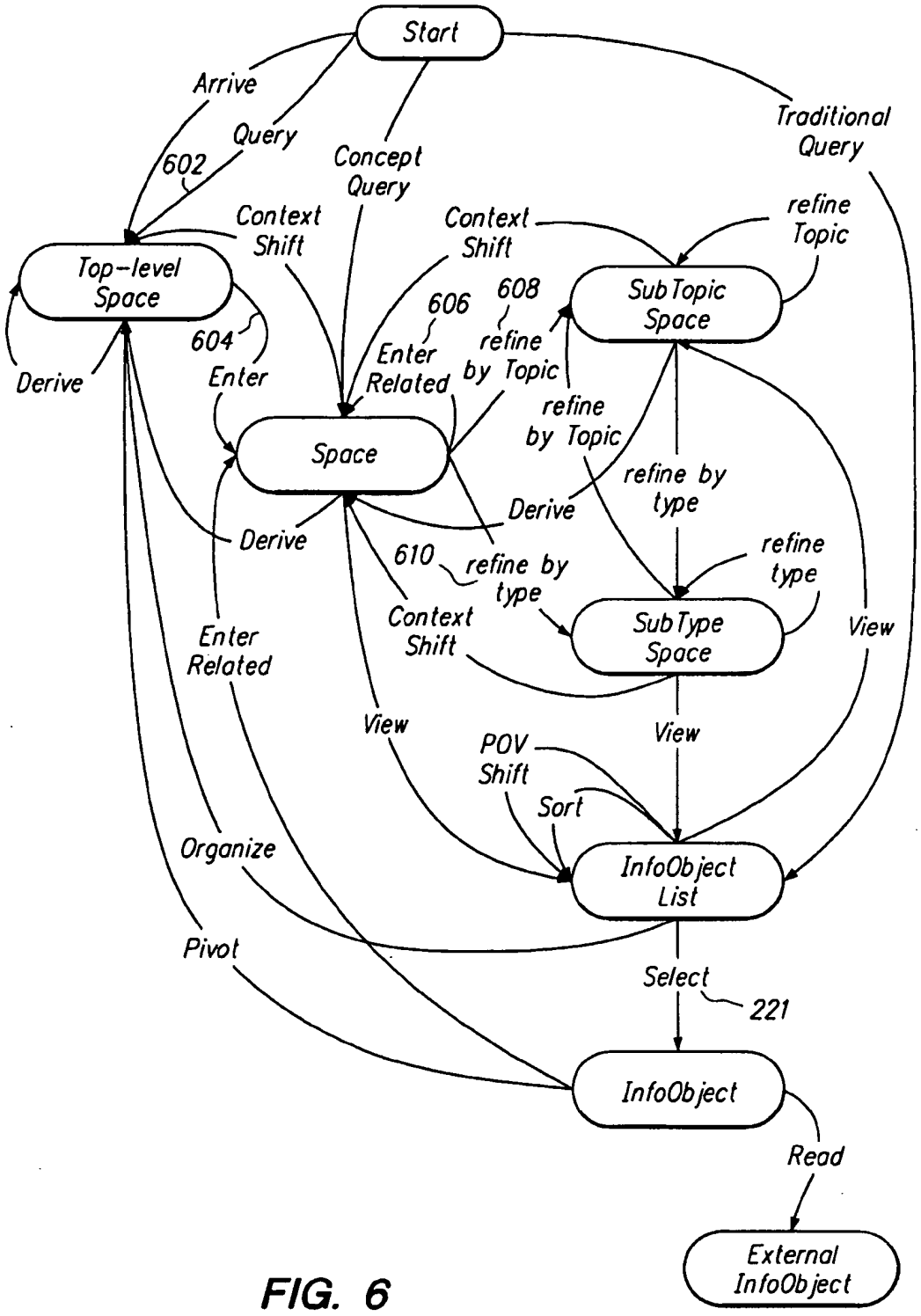

Start
Arrive
Query
602
Concept Query
Traditional Query
Context Shift
Context Shift
refine Topic
Top-level Space
604
606
608
SubTopic Space
Derive
Enter
Enter Related
refine by Topic
refine by Topic
Space
Derive
refine by type
Derive
610
refine by type
refine type
Context Shift
SubType Space
View
Enter Related
View
View
POV Shift
Sort
Organize
InfoObject List
Pivot
Select
221
InfoObject
Read
External InfoObject

FIG. 6

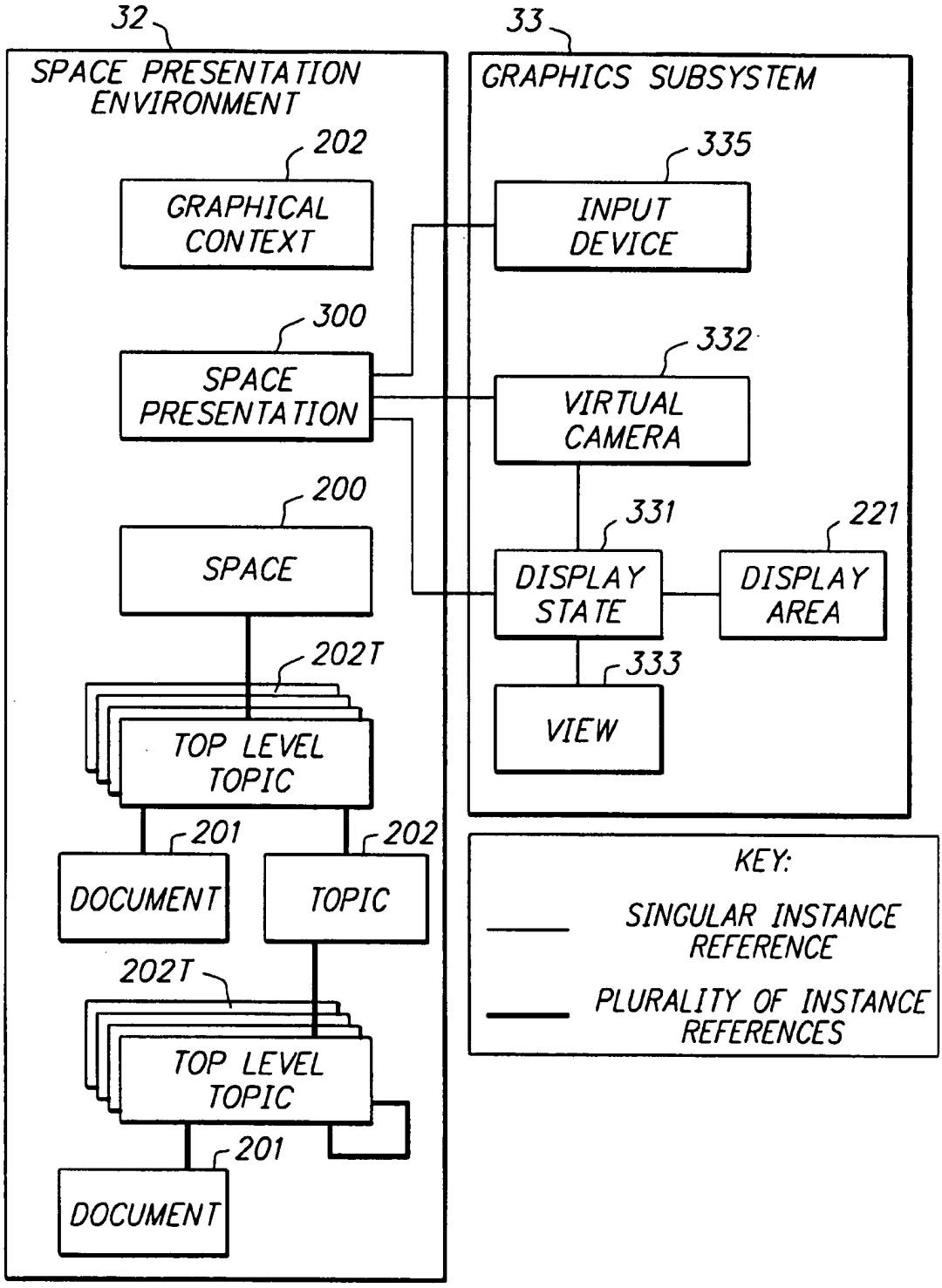

32
SPACE PRESENTATION ENVIRONMENT
202
GRAPHICAL CONTEXT
300
SPACE PRESENTATION
200
SPACE
202T
TOP LEVEL TOPIC
201
DOCUMENT
202
TOPIC
202T
TOP LEVEL TOPIC
201
DOCUMENT
33
GRAPHICS SUBSYSTEM
335
INPUT DEVICE
332
VIRTUAL CAMERA
331
DISPLAY STATE
221
DISPLAY AREA
333
VIEW
KEY:
SINGULAR INSTANCE REFERENCE
PLURALITY OF INSTANCE REFERENCES

FIG. 7

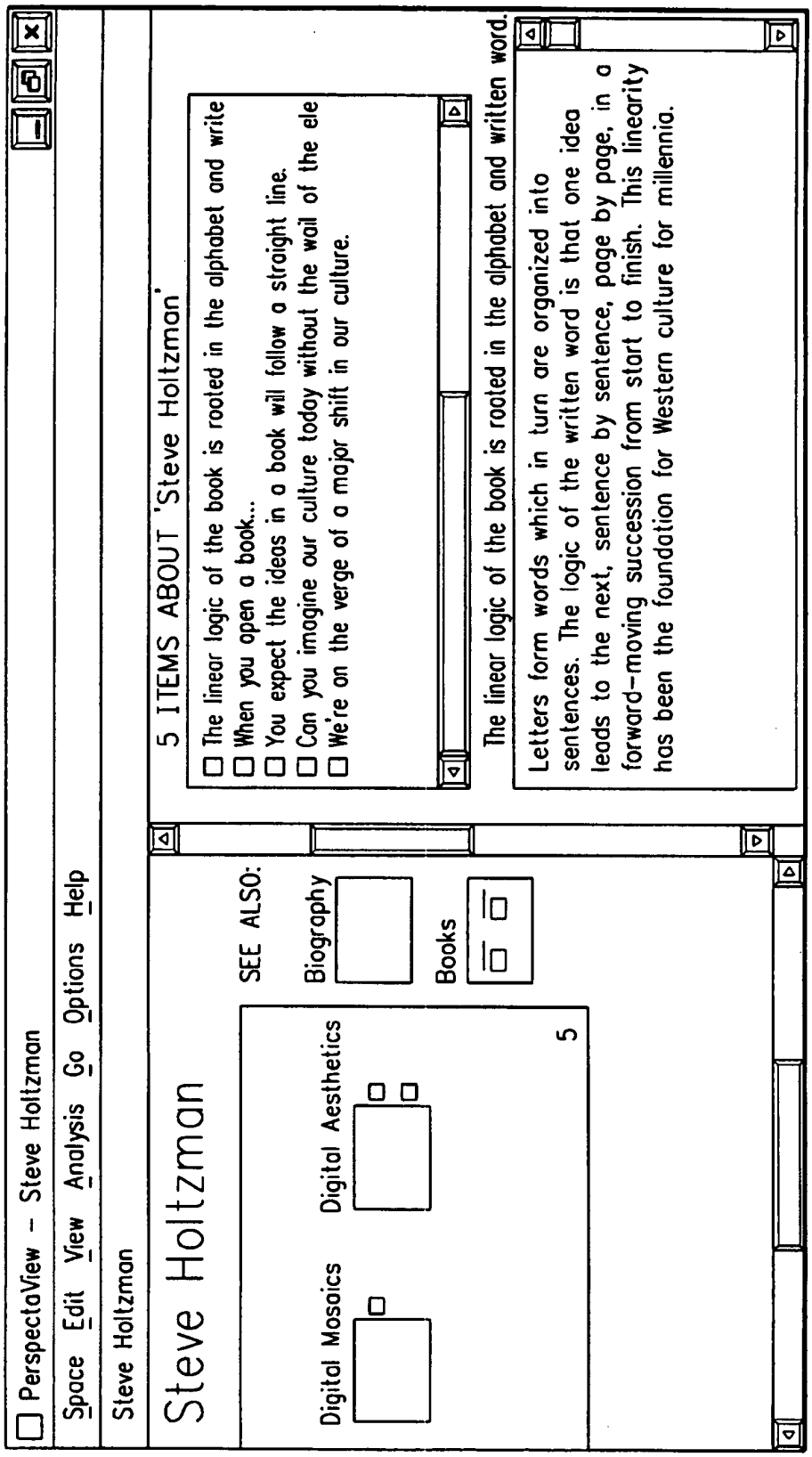

PerspectaView - Steve Holtzman
Space Edit View Analysis Go Options Help
Steve Holtzman
Steve Holtzman
SEE ALSO:
Biography
Books
Digital Mosaics
Digital Aesthetics
5
5 ITEMS ABOUT 'Steve Holtzman'
The linear logic of the book is rooted in the alphabet and write
When you open a book...
You expect the ideas in a book will follow a straight line.
Can you imagine our culture today without the wail of the ele
We're on the verge of a major shift in our culture.
The linear logic of the book is rooted in the alphabet and written word.
Letters form words which in turn are organized into sentences. The logic of the written word is that one idea leads to the next, sentence by sentence, page by page, in a forward-moving succession from start to finish. This linearity has been the foundation for Western culture for millennia.

FIG. 8

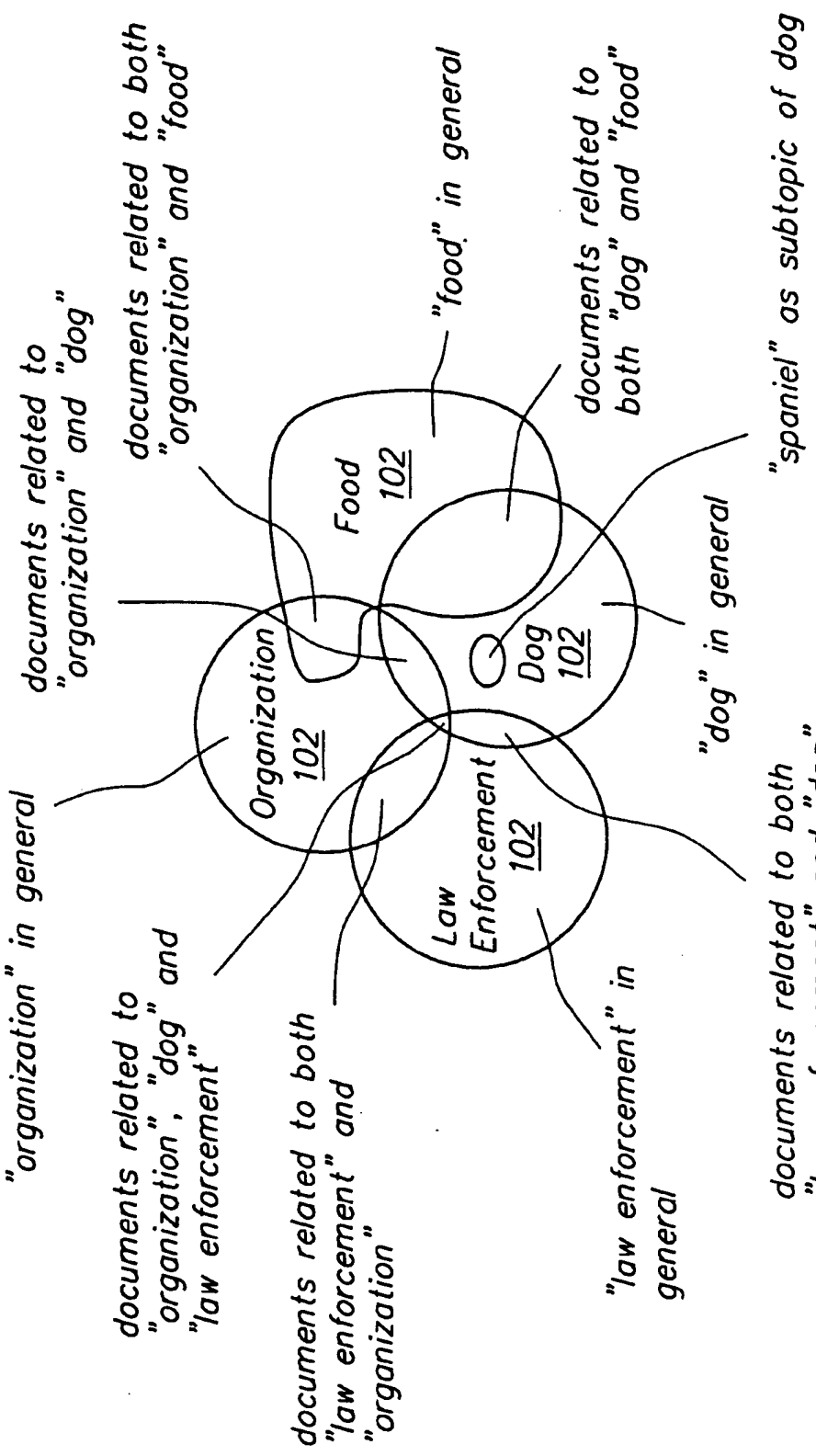

"organization" in general
documents related to "organization" and "dog"
documents related to both "organization" and "food"
"food" in general
documents related to both "dog" and "food"
"spaniel" as subtopic of dog
"dog" in general
documents related to both "law enforcement" and "dog"
"law enforcement" in general
documents related to both "law enforcement" and "organization"
documents related to "organization", "dog" and "law enforcement"
Organization 102
Food 102
Dog 102
Law Enforcement 102

FIG. 9

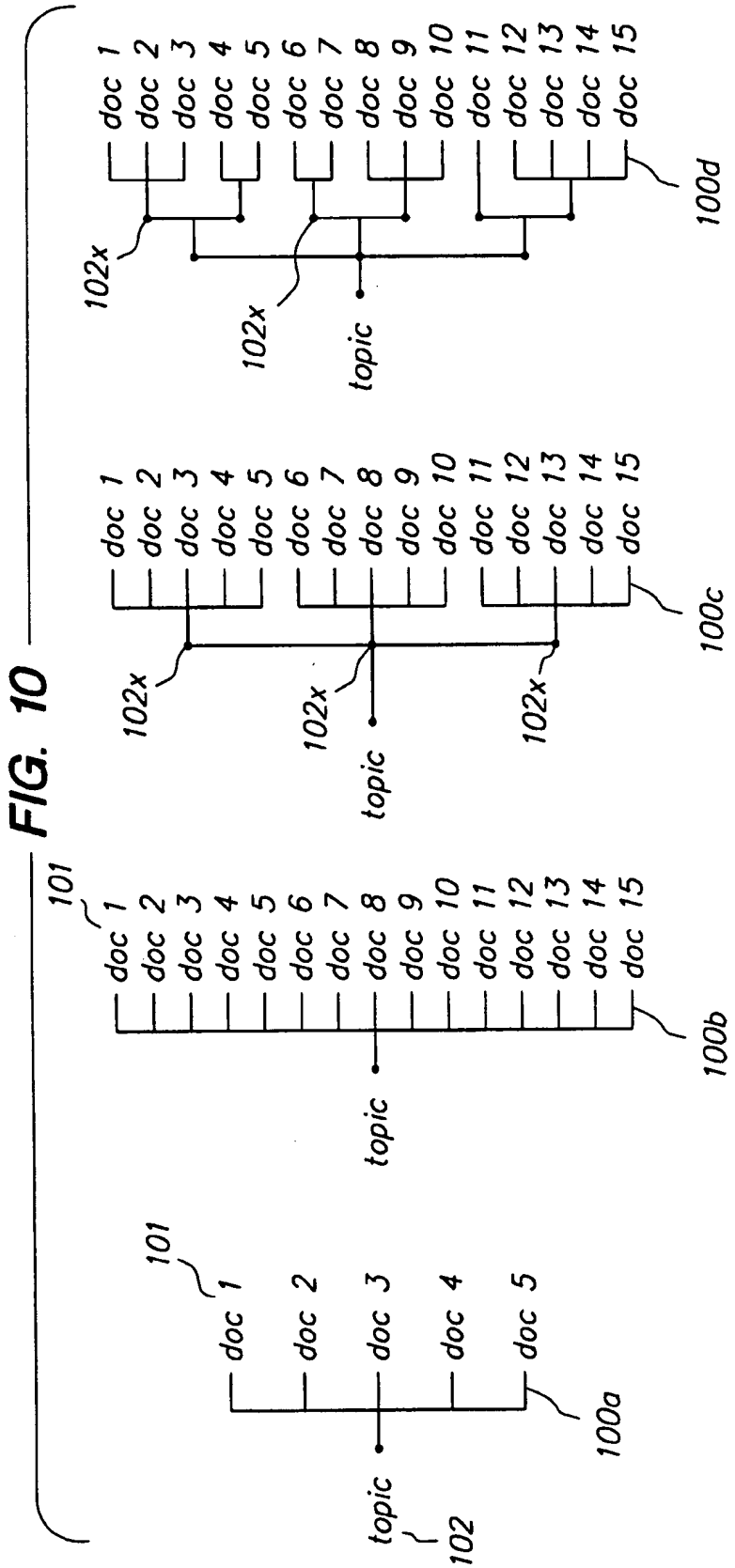

101
doc 1
doc 2
doc 3
doc 4
doc 5
100a
topic
102
101
doc 1
doc 2
doc 3
doc 4
doc 5
doc 6
doc 7
doc 8
doc 9
doc 10
doc 11
doc 12
doc 13
doc 14
doc 15
100b
topic
102x
102x
102x
100c
topic
102x
102x
100d

FIG. 10

IMMERSIVE MOVEMENT-BASED INTERACTION WITH LARGE COMPLEX INFORMATION STRUCTURES

RELATED APPLICATION

This application is a continuation of Serial No. 60/048,150, entitled "Immersive Movement-Based Interaction with Large Complex Information Structures" filed on May 30, 1997 pending, which is incorporated in its entirety by reference herein, and which is assigned to a common assignee as the present application.

MICROFICHE APPENDIX

This application includes a microfiche appendix, including 3 sheets of microfiche and a total of 216 frames.

BACKGROUND

1. Field of the Invention

The invention relates generally to information retrieval systems, and more particularly, to information retrieval systems for retrieving information in multi-dimensional spaces, using 3-D spatial modeling of semantic entities, including topics and documents.

2. Background of the Invention

Most information systems organize documents into some type of information structure, such as a hierarchical, relational or object oriented database. For example, a file system on a personal computer or UNIX workstation consists of files organized into various topic based files, with documents placed into files by topical indexing. The organizational structure of the hierarchies are typically such that a particular document or set of related documents can be found by accessing the appropriate directory or file.

However, when the number of documents in a database grows to be very large or when the contents of the documents are semantically multi-dimensional, that is each document is about many different topics or subjects, a hierarchical organization becomes an inefficient and cumbersome mechanism. This is because a given document may be properly related to a number of different topics; duplicating the topic for storage in many different topic directories dramatically increases storage requirements for the database, and introduces additional problems of maintaining each of the duplicate copies. Further, finding related documents is difficult and time consuming, since a strict hierarchy prevent efficient linking of related topics.

Finding information in these large hierarchies is further exasperated when the organizational structure of the hierarchy is unknown. Many users are unfamiliar with the hierarchical structure of the database they use, and thereby cannot readily identify documents or topics of interest. Current interfaces provide little support for navigating large complex hierarchies in such databases. This is because the user is typically only given a static sense of where in the database or hierarchy they are searching, with no dynamic presentation of such context as the user changes their queries.

What is needed is the ability to control the amount of information in the hierarchy that is presented at any given time. An effective interface would provide the ability to smoothly control the amount of information presented at any given time and the level of granularity of the information. Even still a hierarchical structure is limited in it's ability to express relationships between intermediate nodes and documents.

Another approach to storage and access of documents is to use a relational structure such as that used by relational databases where the fields in the database contains selected attributes about the document, such as the date published, author, so forth. This is an effective technique for storing documents, and is widely used in document databases.

However, accessing documents in a relational database can be difficult, particularly for novice or occasional users. The typical approach to accessing documents in a relational database is to enter a query using a form, which is then processed, and the results returned to the user as a long list of documents that match the query. This works reasonably well only if the documents are tagged clearly and precisely when entered into the database with the appropriate attribute information that users are interested in, and when the number of items returned from the query is less than about twenty. When the number of documents returned is greater than about twenty or thirty the task of finding the document or set of documents of interest becomes increasingly unmanageable, since the user now has to scan or review these documents to determine if there is a more precise query that can be applied to the database. As a result, the user is required to reformulate a query to narrow a search. This is often a difficult task in that the user must match what they are looking for against what is actually available in the database. Since the user typically does not know what is in the database to begin with this can lead to a very frustrating experience.

For example, if a user looking for articles about "Siberian Huskies," she may type an initial query for information about "dogs." The result returned may by a list of hundreds or thousands of articles that were about dogs. A subsequent refinement may be to search for information about "Siberian Huskies" directly, which may result in zero articles. Such a result does not mean that the database does not have any useful information about "Siberian Huskies". Rather, the system likely does have some useful information for the user, only at this point the user does not know this fact, because she does not know how to specify an appropriate query that will result in a useful and manageable amount of information.

An ideal system would provide the ability to start of with a broad search such as "dogs" and analyze the long list of returned documents and place them in an organizational structure that would allow the user to effectively refine the original board query down to a set of documents that may be applicable to her needs. Such a system would have the structure of a hierarchical system to help find information and, the fluidity of an interface to help control how much information is presented at any given time, while simultaneously providing the flexibility to store complex relational information.

A third way of organizing information is by using a hyperlink to connect documents together. In this approach navigable links from one document to another are stored as part of the originating, or source, document. This technique allows users to effectively follow a train of thought as expressed by the author of the document.

One drawback to this approach is that the user is totally at the mercy of the author and his or her selection of which items of information in a source document to provide links for, and to which documents to target those links. If the author fails to link a source document to another document that is related to the source document, the user will not find that other document through the source document. Further, hyperlinks are typically a one-to-one link relationship, i.e.

they only allow one connection from a source to a target document, but not in the opposite direction. Thus, they fail to fully capture the relationship between the documents, and make this relationship fully useful in accessing the documents.

There are several additional problems associated with hyperlinks. One, after the user arrives at the target document, the source context is often lost. These jumps are typically discontinuous requiring the user to re-orient themselves after accessing the target document. For example, if a user is reading a document about Siberian Huskies, and accesses a hyperlinked document about the American Kennel Club, then the context of the source document is lost, and the user is now reading about the AKC. After certain number of these semantic jumps, the user can loose the orientation of where they are in the conceptual structure, and the focus or purpose of their original query. Systems that address these problems would provide some visual cues to help the user maintain context as to where they are in the conceptual structure.

Accordingly, it is desirable to provide an information retrieval system that provides a dynamically constructed respresentation of the context resulting from each query, and which provides this information in a graphical environment where the amount or density of information visually presented to the user is controlled.

SUMMARY OF THE INVENTION

Objectives

The general objectives of the system described in this document include the following:

1. To give users a generalized process for refining a search in a structured and organized fashion;
2. To allow users to see relationships between information such that they can:
   a) understand the contents of a database, and;
   b) more efficiently and effectively access information of interest.

The presentation and interaction model support the objectives broadly defined above by striving for the following objectives:

Provide users with the ability to dynamically control the density and granularity of information within a topic of interest.

Provide users with context at every instance in their interaction with a set of information.

Provide users with an indication of the scope of documents they are working with including both semantic scope and logical topic refinement scope.

Show connections between related topic and documents.

Indicate to users how they can refine their search.

The present invention provides an information retrieval system that overcomes the limitations of conventional systems and satisfies these various objects. In one embodiment of the present invention, a large document collection is segmented into various units of information. In order for these segments of information to be meaningful to the user, the information retrieval system provides three different types of cues to the user: scale, context and an indication of the types of selected relationships between items of information in the information structure.

Scale is the relative measure of semantic abstraction. For example, the concept "golden retriever" is conceptually at a much smaller scale than the concept "animal". The present invention provides a continuous, dynamic, visual representation of the scale of various items of information retrieved in response to a user's query.

Context defines where the user currently is in the information structure relative to her starting point in a session. For example, the user may start off with the topic of "companies," navigate into the subtopic of "software," followed by a subtopic of "databases," followed by another subtopic of "relational," followed by another subtopic of "object." At this point, the user's context will be that of "information about object-relational database software companies." The present invention provides a continuous, dynamic, visual representation of the context of the various items of information retrieved in response to a user's query.

Relationships define how items of information and/or meta-information relate to one another. There are basically two types of relationships: 1) relations between information items (e.g. documents or parts of documents) and meta-information items (e.g. topics); and 2), relations between meta-information items and other meta-information items. These relationships can be grouped together based on the type of relationship. For example, relationships between a topic meta-information item, such as "dogs," and subtypes of dogs, such as "golden retriever" and "black labrador" can be grouped together as subtypes of dogs. Further, information items, or documents, about golden retrievers can be grouped together under the topic meta-information item "golden retriever," where the relation is labeled as "about," and further grouped together under the topic meta-information item of "dogs," with the relation labeled as "subsumed" or "under." These groupings and classifications of groupings of relationships provide a graphically scaleable user interface for dynamically controlling the visual density of information displayed to the user, and provide a mechanism for scaling the size of the document collection without impairing the user's ability to access relevant documents.

Using the foregoing framework, the present invention defines a simple but powerful model of a three dimensional (3D) space that represents an information space. An information space comprises an information structure, the projection of the information structure into a 3D space, a set of reactable 3D graphical objects that populate the 3D space and that represent nodes and relationships between nodes in the information structure, and a virtual camera that represents the user's focus of attention and 3D position in the information structure. At any point in time, the information structure of an information space is dynamically determined in response to a user's query and is a representation of the relationships between a collection of documents that satisfy the query. Such an information structure thereby has scope determined by the user's query. For example, the query "all documents written by Tom Jones from Mar. 1, 1995 to Mar. 1, 1996" defines a specific, dynamically generated information structure.

In the present invention, the 3D information space is the medium through which the user interacts with the information structure to both create queries and see their results. In other words, instead of entering text queries as in conventional systems, the present invention enables the user to create queries by navigating through the 3D information space itself, which is dynamically repopulated with 3D graphical objects representing an information structure which is computed in response to the user's movements (query) in the 3D space

The present invention further provides for a constant density of visual information that is presented to the user. In response to the user's navigation query in an information space, the information retrieval system controls the graphical size of the various 3D objects that will be displayed in response to the user's movements, such that for any display

area, the density of information is constant, where density is a function of the display area, and the number of information items (documents, topics, or other semantic entities) being displayed. This constancy of information density is provided regardless of the semantic scale (whether viewing high semantic scale topics or low semantic scale specific documents) and regardless of the user's context in the information structure.

In one embodiment, the information retrieval system in accordance with the present invention includes a server that dynamically builds and maintains information structures that can be shared among many different client applications, where the information structures are derived automatically from a collection of documents and contain a graph of topics that allow for progressive refinement of the document collection, and a client application that presents information structures to users in a 3D projection of an information space that visually and spatially represents the information structure. The client application preferably includes a camera that represents a user's focus of attention and 3D position in an information space, a set of graphical objects that represent nodes and relationships between nodes of the information structure and have 3D positions in the information space, and a context state that represents the current context of the user in the information space and incorporates the path that the user has taken through the information structure to arrive at the current context. The 3D information space is dynamically generated in response to the user's movements through the 3D information space. The graphical objects are presented by the client application in the 3D information space so that there is constant density of graphical information as the user traverses through the information space.

Preferably, each information space is displayed as a 3D parallel projection space which is infinitely expandable within a normalized volume. A normalized volume is a unit cube.

The set of graphical objects represent semantic entities that are displayed as either text, images, or both in the information space. The graphically objects scale dynamically based on their distance from the camera and their position in the 3D information space, both graphically and semantically. Specifically, in addition to scaling the graphical representation of the object relative to camera position, the graphical objects also scale according to a level of detail of the information associated with the object. The graphical objects are arranged along a z-axis (perpendicular to the surface of the computer display) according to semantic scale. More detailed (i.e. less general or abstract) information appears as the camera zooms along the z-axis in the direction of the computer display (relative to the user).

In one embodiment, the information retrieval system includes a graphical rendering system that scales the graphical objects based on their distance from the camera and their position in an 3D information space, and that supports the dynamic scaling of fonts used to display the graphical objects based on their distance from the camera and their position in the 3D information space.

The information retrieval system operates to provide an interaction model between the user and the system that updates the position of the camera in the 3D information space based on user inputs, updates the positions and level of detail representations of the graphical objects based on the position of the camera in the information space, updates a current context describing the user's present location in and path through an information space including a history of the user's session from their starting point, and allows for both continuous fly-through navigation as well as point-and-click assisted navigation to graphical objects.

In one embodiment, the information retrieval system includes the presentation of cross links in the 3D information space. Cross links show the relationship between a given topic or node in the information structure and the positions of the information objects the node contains as they are positioned in other topics outside of the given topic. Cross links also show the relationship between topics that both contain the same subtopic. Cross links further show the connections between an information object positioned under one topic and all the other topics under which it is also positioned. Cross links are navigable and allow the user to access the topics that are coupled by the cross links.

In yet another embodiment, the present invention is an information retrieval system for displaying information including an information structure having a plurality of semantic entities, such as topics and documents. The information structure is preferably a graph in which the semantic entities are nodes. Each semantic entity has a navigable link to at least one other semantic entity. Each semantic entity is further associated with a graphic object for representing the semantic entity on a display screen. A graphic object may be displayed at any of a plurality of graphic sizes. The system operates with a display window having a variably resizable display area. The display window has an information density which is a function of the number of graphic objects displayed in the display area, such that the greater the number of graphic objects displayed in the display area, the higher the information density.

The system includes a display engine that displays graphic objects of a selected number of semantic entities. The semantic entities are selected from the information structure in response to user queries. The display engine displays the graphic objects of the selected semantic entities but maintains the information density of the display area as a constant at all times, in response to the user's queries for different semantic entities. In this fashion the user has a constant density of information available to view, regardless of the number of semantic entities of the information structure satisfying the user's query.

The present invention provides various methods for displaying semantic information in the form of graphic objects. In one method, there is stored an information structure having a plurality of semantic entities having navigable links to other semantic entities, and graphic objects for representing the semantic entities on a display screen. Each graphic object may be displayed at any of a plurality of graphic sizes. Preferably each graphic object has the same shape. This method includes displaying a first graphic object of a first semantic entity on the display screen, displaying within the shape of the first graphic object the graphic objects of each semantic entity semantically contained within the first semantic entity; and dynamically scaling the graphical size of the displayed graphic objects such that the information density of the display screen is constant.

Another method for displaying semantic information in the form of graphic objects in a display window includes displaying in the display window first graphic objects of a plurality of first semantic entities from an information structure where the display window having a variably resizable display area, and an information density as a function of a number of graphic objects displayed in the display area. A cursor is displayed in the display window, and user inputs are received to move the cursor toward at least one of the displayed first graphic objects. The method includes simulating movement toward a first displayed graphic object by increasing the graphical size of the displayed first graphic objects, and displaying second graphic objects of second

semantic entities contained by the first semantic entities; wherein graphic size of the displayed graphic objects is determined so that the information density of the display window is constant.

In another embodiment a method for displaying semantic information in the form of graphic objects in a display window includes storing an information structure having a plurality of levels of semantic containment, where each level of semantic containment includes a plurality of semantic entities. Each semantic entity includes a navigable link to a plurality of other semantic entities, and is associated with a graphic object for representing the semantic entity on a display screen. The semantic entities in the information structure are such that each semantic entity either semantically contains at least one other semantic entity, is or semantically contained by at least one other semantic entity. In this context, the method includes displaying in the display window graphic objects of at least one semantic entity from an Nth level from the information structure. The display window has a variably resizable display area, and an information density which is a function of a number of graphic objects displayed in the display area.

The method further includes, for each semantic entity from the Nth level that is displayed, displaying in the display window the graphic objects of the semantic entities at the (N+1) level that are semantically contained by the semantic entity from the Nth level.

To access semantic entities in the system, there is displayed a cursor in the display window. The user provides inputs to move the cursor toward at least one of the displayed graphic objects for a semantic entity from the (N+1) th level. In response, the method simulates movement a displayed graphic object of a semantic entity from the (N+1)th level by increasing the graphical size of the displayed graphic objects of the semantic entities from the (N+1)th level, and displaying graphic objects of semantic entities at a (N+2)th level contained by the semantic entities from the (N+1)th level. The method determines graphic size of the graphic objects to be displayed so that the information density of the display window is constant for any number of graphic objects displayed.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** is an illustration of an information structure.

FIG. **2** is another illustration of an information structure.

FIG. **3** is an illustration of an information space in a 3D environment.

FIG. **4** is an illustration of the concept of semantic containment.

FIG. **5** is an illustration of a system in accordance with the present invention.

FIG. **6** is a state transition diagram of the behavior of a system in accordance with the present invention.

FIG. **7** is an illustration of the relationship between the space presentation environment and graphics subsystem.

FIG. **8** is an illustration of a user interface of the present invention showing the 3D space projection of graphic objects with constant information density.

FIG. **9** is an illustration of the logical intersection of topic nodes.

FIG. **10** is an illustration of the dynamic generation of topic nodes to maintain constant information density.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

I. Definition of an Information Structure

FIG. **1** represents an information structure. An information structure **100** is an organized graph structure that represents the relationships between a plurality of document nodes **101** and topic nodes **102**. (Throughout this disclosure references to the term "node" without any subsequent reference number are understood to reference to any type of node.) Each node can be both a parent or a child node.

Each document node **101** in the information structure **100** has a set of attributes (such as the author, date, title, and the like) and is "about" a set of topics that are represented by the topic nodes **102**. Each document node **101** is preferably described by the following items:

title
summary
author
creation and publication dates
source of the document

A title is a short textual description of the document. A summary is a more lengthy description of the document contents. Optionally, the document node **101** may contain either a reference to the location of the original document (such as a uniform resource locator (URL) hyperlink reference), or the actual contents of the document.

As described above, documents and topics may have various relationships to each other. Relationships between the plurality of document nodes **101** are represented by a plurality of interconnected topic nodes **102** that represent attributes of the document nodes **101**, and a plurality of topic link relationships **103** between topic nodes **102**. A topic node **102** represents either 1) a single concept (e.g. "computer music" or "music computer"), or 2) a logical combination of a plurality of concepts (e.g. "'computers AND 'music'"). Each topic node **102** is preferably represented by the following items

textual label
short textual label
gloss

The textual label is a complete textual description of the topic node **102**, such as "'computers' AND 'music'". The short textual label is a textual description of the topic node **102** relative to it's parent topic node **102**. For example, if a parent topic node **102***a* is labeled "computers" and a child topic node **102** represents a logical intersection between "computers" and "music", then the child topic node **102** would have a short label of "music". FIG. **9** illustrates another example of the logical intersection of topic nodes, here with the various intersection of the topics "law enforcement," "dog," "food," and "organization." The figure illustrates that the topic "dog" contains or encompasses all documents which in any way are associated with "dog", irrespective of whether or not they are subtype relationships.

The gloss contains a full textual description of the topic node **102**. For example, for a topic node **102** labeled "computer", the gloss might be "An electric machine that performs high-speed mathematical or logical calculations or that assembles, stores, correlates, or otherwise processes and prints information derived from coded data in accordance with a predetermined program".

Topic link relationships **103** generally fall into two classes: parent-child relationships and sibling or related relationships The parent-child relationships define a plurality of levels in the information structure. The parent-child relationships between topic nodes **102** described above fall

into two broad categories: semantic subtype relationships **103**, and intersection relationships **103**. A semantic subtype relationship **103** is a relationship that is a conceptual relationship between two topics, such as one topic being conceptually broader and including a second, narrower topic. For example, the topic "dog" is conceptually related to the topic "mammal," and would be a subtype of it.

An intersection relationship **103***c* is where a topic node **102** represents the logical AND of two or more topics **102**.

Nodes at the same level, and nodes at different levels may be interconnected by related topic links **103***d*. Whereas conventional hierarchical structures only represent parent-child relationships, the graphical nature of the information structure **100** handles more complexity than a fixed hierarchical structure and contains information about the related topic links **103***d* between nodes in the structure.

The information structure **100** of FIG. **1** is merely exemplary. In an actual embodiment, the information structure **100** would be significantly larger, perhaps including hundreds or thousands of nodes. The relationships between the nodes however, have the described structural features.

In the structure represented in FIG. **1**, semantic entities called "topics" are represented by the topic nodes **102**. For each topic node **102**, there also exists a document link set **110** of document links **111** to document nodes **101** that are "about" the topic represented by the topic node **102** The document nodes **101** may be "about" many different topics, and hence, there may be a plurality of document links **111** from a plurality of topic nodes **102** to a single document node **101**. This plurality of document links **111** relationships captures the semantic multi-dimensionality of the information structure **100**, enabling the same document to be related to many different topic nodes **102** within the information structure so long as the document node **102** for the document is about the same topics that the topic node **102** represents.

Referring also to FIG. **2**, the topic nodes **102** of the information structure **100** each have a measure of relative semantic scale **219**, where the semantic scale is reduced or narrowed by moving 'down' the graph via child relationships. Hence, topic nodes **102** that are 'lower' in the information structure depicted in FIG. **1** or FIG. **2** are narrower or lower in semantic scale. A node higher up in the structure is more general, or at a conceptually higher semantic scale than a node on a branch lower down in the structure. In FIGS. **1** and **2**, the information structure **100** is presented so that the vertical placement of nodes represents both the relative semantic scale, and the relative semantic containment of topics in the information structure **100**, though in fact there is no vertical organization inherently present in the information structure **100**. By way of specific example, a node **102** may represent the topic "dogs", which has a higher, or more general degree of semantic scale than a child node **102** which represents the topic "Siberian Huskies."

For every information structure **100** there exists a singular root node **102***r* that contains a plurality of root topic link relationships **103***r*. These root topic link relationships **103***r* define a set of top-level topic nodes **102***t* that are the entry points into the information structure **100**. Traversal of the information structure **100** is initiated through these nodes **102***t*. Related links **105** between nodes under different topics express the conceptual relationships between various topics to each other.

The meaning of each topic node **102** in the information structure **100** is dependent upon the path from the root topic node **102***r* to the given topic node **102**. This traversal path is called a context **107** and consists of the root topic node **102***r* followed by a series of topic link **103**/topic node **102** pairs to the given (or current) topic node **102**. This context **107** is useful for a user to interpret both the topic nodes **102** and the document nodes **101** for the current topic node **102**. The context **107** is also useful for giving the user a global understanding of where she is in the information structure **100** and what her current location in the information structure **100** means.

Given the information structure **100** as described above, a set of cross links **105** represents cross relationships between document nodes **101** in a single topic node's document link set **110** and the same document node **101** contained in all another topic node's document link set **110**. These cross links **105** define the additional contexts **107** in which each of the document nodes **101** assigned to a topic node **102** also exist.

Semantic containment is a function of parent-child relationships in the information structure. Referring to FIG. **2**, a parent node **102** is defined to semantically contain all and only its child nodes **102**, including any further grandchild nodes **102**. For example, node **102***a* semantically contains nodes **102***b* and **102***c*, and their respective child does, **102***u*, *j*, *k*, *l*, but does not semantically contain nodes **102***e*, **102***f*, and **102***g*. Semantic containment is used to express the relationships of topics to subtopics in the information structure **100**. Further, the number of child nodes **102** for any given topic node **102** is limited to the number of topic nodes **102** that can be displayed on a screen at any given time so that the density of information can be maintained. During the structure generation process, when the number of child nodes **102** of a topic node **102** grows too large, the set of child nodes **102** is broken down into groups where the groups are represented by a new set of child nodes **102**. This process is also applied to the relationship between document nodes **101** and topic nodes **102**.

As illustrated in FIG. **10**, when the number of document nodes **101** of a topic node **102** leaf of the information structure **100** is above a user specified threshold, a new set of topic nodes **102***x* is constructed to break down the set of document nodes **101** into smaller groups. In FIG. **10**, the information structure **100***a* has only 5 document nodes **101**, and thus is easily reviewed by the user. However, information structure **100***b* has 15 document nodes **101**, which is too many for a user to easily review. Accordingly, a new set of topic nodes **102***x* is dynamically created, as shown in information structure **100***c*. If the user threshold is less then 5, for example, then further intermediate topic nodes **102***x* can be generated within the information structure, as shown in information structure **100***d*. This process of breaking down larger groups into smaller groups and labeling the groups based on common attributes enables the present invention to present a constant density of information to the user at any given point of time regardless of the number of documents or topics which currently satisfy a user's query, and allows users to more specifically narrow their search. This process is also the basis of the information structure generation function.

Appendix A provides a description of one embodiment for the information structures using topic nodes in a knowledge base, and a structuring process for generating the information structures in response to a query.

II. Definition of a 3D Information Space

To provide a means of navigating through the large collection of documents represented by information structure **100**, the present invention defines an immersive 3D information space **200** that represents the organizational properties of the information structure **100**. This 3D information space **200** is the graphical representation of the

information structure **100** that is visually represented on a 2D computer display screen **220**. The 3D information space **200** mirrors the information structure **100**, adding graphical representations of the underlying information structure **100**. In other words, for each element in the information structure **100** there exists an element in the information space **200** that contains graphical representations of the information structure **100** element.

Referring to FIG. **3** and to FIG. **7**, the display area **221** of a computer display screen **220** is defined to represent the X and Y axes of a 3D coordinate system **230**. The Z-axis **232** is positioned perpendicular to the plane of the display area **221**. The Z-axis **232** is used to represent the semantic scale of the information structure **100** and the semantic containment relationships between nodes. The XY plane **233** is used to represent different types of sibling relationships at any particular level of the information structure **100**.

A 3D information space **200** has a base 3D coordinate system **230** that represents the root topic node **102***r* of the information structure **100**. The set of top level graphical topics node sub-spaces **202** that represent the top-level topic nodes **102***t* in the information structure **100** are defined in the base 3D coordinate system **230**. Each topic node **102** in the information structure **100** is represented by a graphical topic node sub-space **202** (or more simply a graphical topic node **202**). Each graphical topic node sub-space **202** has an 3D location defined in a local 3D coordinate system **231** that is relative to its parent graphical topic node sub-space **202**. Each parent graphical topic node sub-space **202** contains all of the child graphical topic sub-spaces **202** that are represented by the child topic nodes **102** of the corresponding parent topic node **102** for the given parent graphical topic node sub-space **202**.

In order to present the information structure **100** to the user on a computer display, each graphical topic subspace **202** in the information space **200** is represented by a 3D topic graphical object **242** that is sensitive to the movement of a virtual camera **332** in the information space **200**. To maintain the constancy of information density, the graphical size and presentation of the topic graphical object **242** changes depending upon the amount of display area **221** available and the number of topic graphical objects **242** available in the given context **107**. The topic link relationships **103** between topic nodes **102** can optionally be represented by a 3D line **243** that connects between the 3D anchor points of the two graphical topic sub-spaces **202**. Optionally, the semantic containment that is represented by the parent-child relationship in the information structure **100** can be represented by a graphical container object that bounds the topic graphical sub-space **202**. An example of a graphical container object is a 3D box that encompasses all of the child graphical sub-spaces **202***b*.

Within the information space **200** there are a set of graphical document objects **201** that represent the document nodes **101** Since the each document node **101** may be linked to multiple topic nodes **102**, graphical document objects **201** may have multiple locations within the 3D coordinate system **230** where each location is defined in the local 3D coordinate system **231** of the graphical topic sub-space **202**. The present invention defines two different ways of handling the multi-dimensionality of these relationships. One, for each document node **101** location in the information structure **100** there can be a multiple graphical document objects **201**, each with a 3D location **235** defined in world 3D coordinates **230**. This approach can also be modeled by having one graphical document object **201** that is graphically drawn in multiple 3D locations where each location is defined relative to the local 3D coordinate system **231** of each of the graphical topic sub-spaces **202**.

The second approach is to use a single graphical document object for each document node **101** in the information structure **100** and have its 3D location **235** move/animate from one position to another in the 3D coordinate system **230** to another.

As described above, the context **107** represents where the user is in the information structure **100** relative to the root topic node **102***r*. In the corresponding information space **200**, a current graphical context **209** is determined by the particular graphical topic node sub-space **202** that user's location currently occupies, and the path traversed through the information structure **100** to reach the topic node **100** that the graphical topic node sub-space represents. As a result, each graphical topic node subspace **202** in the information space **200** corresponds to a unique context **107**. This context **107** is graphically represented by a singular graphical context **207** (not shown). The graphical context **207** consists of a textual or pictorial representation of the context **107**. The graphical context **207** is displayed in one of seven locations in the 2D display area **221**: center, top left corner, top center, top right corner, bottom left corner, bottom center, and bottom right corner. Previous contexts (including both context **107** and its corresponding graphical context **207**) are maintained in a context stack. The way graphical topic nodes **202** are added to the current graphical context **207** (and hence the context **107** which operates in parallel to graphical context **207**) depends on their relationship to the previous topic on the context stack **107**. When a child node **102** is added to a context **107** it is appended to the current context **107**. When the user moves into a sibling node **102**, the context **107** is changed to represent the path to the sibling node **102**. As noted above, the changes to context **107** are reflected directly by the graphical context **207**.

In the present invention, information retrieval and query formation are controlled by movement through the information space **200** from one graphical topic node **202** to another. The movement of the user through the information space **200** from a graphical parent node **202** to a graphical child node **202** is interpreted as either 1) a semantic refinement of the parent topic node that narrows the semantic scope, or 2) a Boolean query that AND's the topics **102** associated with the two nodes **202**.

For example, if a user moves from a graphical topic node **202** of "pets" into the child graphical node **202** "dogs" that is linked by a subtype topic link, this is interpreted as a semantic refinement of "pets." If a user moves from a graphical topic node **202** of "pets" into the child graphical topic node **202** "dog food" that is linked by an intersection topic link, the Boolean description of the current context **207** resulting from that movement may be described as "Pets AND Dogs". Repeated movement from parent to child graphical topic nodes thus creates successive Boolean logical operations. As a result, successive movement from parent to child graphical topic nodes **202** in the information space **200** constrains the document set **110** in the information structure **100** that is associated with the current context **107**.

Alternatively, if a user moves from a graphical topic node to a sibling graphical topic node via a related topic link **203** (that represents a related topic link **103**), the sibling node replaces the previous node on the graphical context stack **207**. For example, if the user moves from the graphical topic "Dogs" to the sibling graphical topic "Cats", "Cats" would replace "Dogs" on the graphical context stack **207**; the corresponding replacement occurs in the context stack **107**.

Thus, movement in the information space **200** defines both the query to the information structure **100**, and the resulting display of the information space **200** which is updated to reflect such movement.

FIG. **4** depicts the principle that some topic nodes **202.1** are semantically contained by a central topic node **202.0**, while other topic nodes **202.2** are merely connected to it. In the 3D graphical environment of the present invention, graphical topic nodes **202.1** contained by another graphical topic node **202.0** are represented as being further forwards (into the display area **221**) along the Z-axis **231**. Graphical topic nodes **202.2** which are connected to a central graphical topic node **202.0** are arranged on the display area **221** of the screen in the XY plane **232** containing the central graphical topic node **202.0**. In addition, movement forward on the containment or Z-axis **231** results in further concept refinement, reducing the number of relevant documents, while movement to those topics which are connected in the XY plane **232** do not necessarily narrow the search space.

Movement forward along the Z-axis **231** results in a zoom operation that stretches the XY plane **232** around the user's position in 3D coordinates **230**. As a result the graphical objects (topic nodes **202** and other objects) move apart and grow in size on the 2D display area **221** allowing graphical objects that correspond to lower level nodes in the information structure **100** to be presented. This zoom operation reveals more details around the center of the 2D display area **221**. Movement backward along the Z-axis **231** shrinks the XY plane **232** around the user's position in 3D coordinates **230**. As a result of the XY plane **232** shrinking, graphical objects move closer together.

As either of these movements along the Z-axis **231** take place, the information retrieval system selects which graphical objects to display and their size so that the density of graphical information is constant. For example, as the user moves backward on the Z-axis **231**, the graphical objects **202** that correspond to lower level nodes **102** in the information structure **100** are removed from the display area **221**, and higher level nodes **202** are added. The direct result of this process is the presentation of a constant level of information density controlled directly by the user's movement in the information space **200**.

III. System Description

A. High-level System Description

Referring to FIG. **5**, there is shown an embodiment of an information retrieval system **500** in accordance with the present invention. The system **500** consists of three main components:

A database **10**;

An information structure server **20**; and

A information space presentation and interaction client **30**.

The database **10** stores a collection of documents **101**. For each document **101**, the database **10** stores a set of meta-data that describes the document **101**. This meta-data comprises: a) a set of attributes that describe the document **101** (e.g. who the author was, when the document was created, etc.), and b) a set of topics **102** define what the document is "about" (e.g. "Siberian Husky," "dogs," "dog food"). The derivation of the document attributes and topics **102** that describe a document **101** is not important for this disclosure, only that they exist and can be stored in the database **10**.

The information structure server **20** is responsible for responding to user queries **50** for information structures **100** and delivering the information structure **100** back to the information space presentation and interaction client **30**. In turn, the information space presentation and interaction client **30** is responsible for presenting the information structure **100** to the user in the form of an information space **200**. In addition, the client **30** is also responsible for responding to a user's interaction with graphical objects of the information space **200** presented to the user and dynamically reorganizing the graphical presentation of the information structure **100**.

As illustrated in FIG. **5**, the three major components are broken down into the following primary components:

Discourse Manager **31**;

Space Presentation Environment **32**;

Graphics Subsystem **33**;

Data Manager **34**;

Request Broker and Object Transport Mechanism **35**; and

Dynamic Content Organizer **21**.

The microfiche appendix contains an exemplary embodiment of class definitions for an implementation of these components.

The discourse manager **31** is responsible for directing the high level interaction between the user and the system **500**. The discourse manager **31** interfaces with the request broker **35** to query the information structure server **20** to generate an information structure **100**. Once the discourse manager **31** has received the initial elements of the information structure **100**, it directs the space presentation environment **32** to present the information space **200** to the user.

The space presentation environment **32** is responsible for coordinating the fine grained interaction between the user and an information space **200**. Referring to FIG. **7**, the primary mechanism for controlling this interaction is a space presentation **300**. The space presentation **300** presents the graphical objects to the user on the 2D computer display screen **220** using the graphics subsystem **33**. The graphics subsystem is responsible for displaying graphical objects (e.g. graphical topic nodes and documents **201**) on the 2D computer display screen **220**. The user, represented by a virtual camera **332** situated in the information space's 3D coordinate system **230**, can then respond by moving a cursor **222** (FIG. **2**) relative to the graphical topic nodes **202** and graphical document objects **201** shown on the 2D computer display screen and, using an input device, move his or her position around the information space **200**. When a user moves close to a graphical topic node **202** (where "close" is defined by a set of rules attached to the graphical topic node **202**) in the information space **200**, an event is triggered that results in the space presentation environment **32** querying the data manager **34** for more elements of the information structure **100**. One embodiment of a visual camera **332** is disclosed in Appendix B.

The data manager **34** is responsible for managing the information structure **100**. Since the information structure **100** is inherently a graph and can contain cyclic references to nodes in the graph, the data manager **34** provides functions to facilitate referencing cyclic graph nodes in the information structure **100**. This is accomplished via a single level of indirection through a central object reference table maintained by the data manager **34**. The data manager **34** also provides a mechanism to allow those cyclic graphs to be transported across the network from the information structure server **20** to the information space presentation and interaction client **30**. The data manager **21** acts as an intermediary between the space presentation environment **32** and the dynamic content organizer **21**, and provides a transparent mechanism for the space presentation environment **32** to query the information space server **20** for more details of an information structure **100**. The data manager **34** component is shared between the client **30** and the server **20**.

The request broker and object transport mechanism **35** provides low level network communications between the client **30** and the server **20**.

The dynamic content organizer **21** is responsible for responding to user queries and generating information structures **100**, A user query is typically interpreted into the form of a Standard Query Language (SQL) select statement **22**. This select statement is used to select a set of documents **101** and their corresponding meta-data (including a set of topics **102** associated to each document **101**) from the database **10**. The document **101** meta-data is used to construct an information structure **100** with the properties described above.

The dynamic content organizer (DCO) **21** generates an information structure **100** by first selecting all of the documents **101** from the database **10** that match the select statement **22**. For each of these documents **101** the DCO **21** selects the corresponding set of document topics **102** that describe what the document **101** is about. From these sets of document **101**-topic **102** co-occurrences, the DCO **21** constructs a co-occurrence graph that describes the relationships between the document topics **102** as defined by the documents **101**. After creating this graph, the DCO **21** traverses the graph to find the top-level topic nodes **102***t* and places them under a newly constructed a root topic node **102***r*.

The method described above is one way that information structures can be generated. Appendix A describes this method in further detail. Other methods can be used to generate an information structure **100**, but once they are in the form for an information structure **100** described above this disclosure describes how the user can interact with those information structures **100**.

B. Behaviors of Objects with User Interaction

The system **500** provides two basic phases of interaction with the user: 1) initiation of the information space presentation, and 2) dynamic presentation and interaction with an information space. FIG. **6** illustrates a state transition diagram for the behavior of the system.

1. Initiation of the Information Space Presentation

The interaction with the information space **200** is initiated by the user entering **602** a query for information (in the form of a Standard Query Language select statement **22**). The discourse manager **31** receives this query from the user and relays it to the dynamic content organizer **21** via the request broker **35** to get an initial information structure **100**. From this information structure **100**, the discourse manager **31** constructs an information space **200** that it passes to the space presentation environment **32** for presentation **604** to the user. When the user requests a new information space **200** (e.g. by entering **606** a related node's subspace **202**, by refining by topic **608**, or by refining by type **610**), the discourse manager **31** is responsible for taking the current information space **200**, stopping the current presentation, pushing it onto a space presentation stack, and then loading the new information space **200**. The space presentation stack maintains the users history of queries **22** and resulting information spaces **200** over an interaction session with the user.

While the discourse manager **31** controls the course grained interaction with the user, the space presentation environment **32** handles all of the fine grained user interactions within an information space **200**. The space presentation environment **32** is responsible for presenting the initial set of graphical objects that represent the information space **200** to the user, and then allowing the user to move relative to the graphical objects positioned in the information space **200**'s 3D coordinate system **230** and dynamically trigger queries **602** for further refinements or generalizations within the information structure **100**. For example, when a user moves forward along the Z-axis **231** relative to a graphical topic node **202** and crosses into the graphical topic node sub-subspace **202**, the space presentation environment **32** detects this event and dynamically queries the data manager **34** for additional topic nodes **102** in the information structure **100**. If the user has not traversed into this portion of the information structure **100** previously, the data manager **34** issues a request to the information structure server **20** to generate additional levels of the information structure **100** as appropriate for the given context **107**. The request broker **35** relays this request between the client **30** and the information structure server **20**.

2. Dynamic Presentation and Interaction with an Information Space

FIG. **7** illustrates the relationships between the space presentation environment **32** components and the graphics subsystem **33** components. The primary component for driving the presentation of and interaction with an information space **200** is a space presentation **300**. The user's interaction with the information space **200** is initiated by the space presentation **300** traversing the information space graphics structure **200** and issuing requests to the display state **331** to draw the graphical objects (e.g. graphical topics **202** and document objects **201**) on the 2D display screen as represented by the display area **221**. To perform this display operation, the display state **331** uses a virtual camera **332** that models the user's position in a 3D coordinate system **230** and a view **333** that maps the 3D coordinate system **230** into the 2D coordinates of the 2D display area **221**. In the model preferred in the present invention, the view **333** is used to zoom, or expand the 3D coordinate system **230** as the user moves forward along the Z-axis **231**, resulting in objects moving further apart in the 2D display area **221**. As the 2D display area **221** expands, the size, selection and amount graphical information displayed on the 2D display screen is changed to ensure that the density of graphical objects displayed remains constant, though the selection of nodes in the information structure **100** changes. This process is described in more detail below.

After the initial presentation of the information space **200** and graphical objects therein to the user, subsequent interaction with the information space **200** occurs in four phases:

1. User movement;
2. Space presentation **300** reaction;
3. Space presentation **300** animation update; and
4. Space presentation **300** display.

In the first phase, user movement, the user moves his or her position in the space by moving the virtual camera **332**. Movement of the virtual camera **332** is controlled by the space presentation **300**. The space presentation **300** receives input events from the input device **335**, such as mouse movements, mouse button clicks, and keyboard events. The space presentation **300** maps these events to movements of the virtual camera **332** in the 3D coordinate system **230**.

The second phase is to react to the user's movement in the information space **200**. This reaction phase is also executed by the space presentation **300**. The space presentation **300** traverses all of the topic nodes **202** and documents objects **201**, represented as graphical objects in the information space **200**, and they in turn react to the user's 3D position relative to that node, where the user's position is defined by the virtual camera **332**. The reaction of each node is controlled by its parent node. This gives global control of the reaction within the context of each of the topic nodes **202**.

The reaction of the nodes is defined by a set of rules that are based on the user's relative position to a graphical object

and the current state of the system. For example, if the position of the virtual camera **332** moves into the subspace defined by the graphical topic node **202**, the graphical topic node responds by presenting its child nodes **202**. If a graphical representation of the child nodes **202** does not exist, the graphical topic node will request the corresponding child topic nodes **102** from the data manager **34** that manages the information structure **100** and will dynamically build the corresponding graphical topic nodes **202**. If the data manager **34** does not have the requested topic nodes **102** in the memory of the client software **30**, a request is issued to the information structure server **20** to build the corresponding information structure **100**'s topic nodes **102**. Other actions that can occur in the react phase include:

i) setting the target state of graphical objects, e.g. the target size, color and/or location of an information object.

ii) a change in global state, such as the change in graphical context **207** (which is reflected in the underlying context **107**).

iii) pushing the previous graphical context **207** onto the context stack.

iv) a change in local state, e.g. setting which node in a sub-graph is active.

Following the react phase is the animation update phase. During the animation update phase all of the graphical objects whose current state (e.g. current location in 3D coordinates **230**) is different from the target state that was set during the react phase are incrementally updated to a new value that that is closer to the target state; or, if the current state is close enough to the target state, the current state is set to the target state. The incremental updates from the current state to the target state can follow linear rate of change or can be specified by a function of either time or a normalized value between 0.0 and 1.0.

The final phase is the display phase. In the display phase, the structure of graphical objects is traversed and displayed depending upon their current state. A 3D parallel projected information space **200** with perspective scaling is achieved by computing the position and size of the graphical objects based on the user's position in the information space **200**, as represented by the virtual camera **332**, and distance of the graphical objects from the user as described above. In this display process, the scale of the graphical objects changes as a function of the position of the virtual camera **332** relative to the graphical objects and the mapping specified by the view **333**. This scale function is applied by the display state **331**. Using this scale function the space presentation **300** determines how much information is presented to the user such that the display area **221** maintains a constant amount of information density. FIG. **8** illustrates a sampler user interface **800** showing the 3-D space projection of the information space **200**, and graphical objects with constant information density.

Appendix A

Dynamic Topic-Based Organization of Documents

I. Abstract

A system for organizing large document sets by creating graphs (tangled hierarchies) of Topic nodes is presented. Each Topic represents a subject (described as a set of concepts) and has a set of documents attached to it that are "about" the subject. Topics are connected via links that represent the semantic relationships between their subjects, such as conceptual generalization, refinement, and association (non-hierarchical "sideways") relations. The graph of Topics is created dynamically in response to a user query and a set of documents.

The graph has the property that for each Topic, a set of connected sub-Topics provides coverage over and distinction between the set of documents about the Topic. Thus, moving "downwards" (from Topics to sub-Topics) recursively in the graph is effectively refining a query: the sub-Topic is a more finely articulated conceptual description of the subject of interest, and there are fewer documents that "match" the sub-Topic.

II. Problem

As more information becomes available on-line, the problem of searching through it to find specific information is intensified. There are two general aspects of this problem:

finding a relevant document (or smaller set of documents)

understanding what a set of documents is "about" (a summary)

One approach to this problem uses traditional Information Retrieval methods (as at TREC) that focus on maximizing "recall and precision": return all and only the relevant documents to a query. The success of IR systems varies widely, usually performing acceptably only with restricted types (domain and style) of documents and with large queries (e.g. an example document as a query). Performance degrades considerably in the "search engine" case, where a query may be only a few words, and the set of documents is unrestricted (e.g. the entire WWW). In this case, large unmanageable lists of tens of thousands of irrelevant documents are usually returned. In addition, their is no mechanism for representing a summary of a set of documents.

A second approach (e.g. Yahoo) uses static Topic hierarchies as "containers" of documents: all documents are tagged (by humans or superficial text analysis) with labels that indicate what Topic category they belong in; users navigate through the hierarchy to find a particular Topic of interest and the associated documents. The problem with this approach is that the hierarchy is the same regardless of the user's query and the document set. Thus, it does not represent the document set (summary), and it is very inefficient for a user to find what they are looking for (they must manually trudge through the same hierarchy of Topics every time). Further, the system is limited in its applicability to those subject categories which its single hierarchy covers; it cannot dynamically leverage other sources of hierarchical information as needed.

A third approach uses statistical clustering algorithms to automatically group related documents together, thus avoiding the problems of static hierarchies that do not change for different queries and document sets. The problem with these systems is that the strong semantic relations between Topics (e.g. those in static hierarchy systems) which are useful for navigation are lost. Thus, there is no clear sense of semantic refinement between Topics, nor is it possible to know how different Topics are related; the result is that the interface to such a system is less consistent and intuitive.

III. Architecture

The system uses a combination of these three approaches to create interconnected Topic hierarchies dynamically in response to a user query and a document set. The preferred approach is implemented as follows:

A system that includes:

a) a database of documents tagged through Linguistic Analysis;

b) a Knowledge Base (KB); and

c) a Structuring Process.

A client interacts with the system to provide user queries and output selected documents and topics.

A. Linguistic Analysis

1. Overview of Linguistic Analysis

Linguistic Analysis produces several Topic labels that represent what each documents is "about". Linguistic Analysis includes:

a) tokenizing, morphological processing, part of speech tagging, and phrase level parsing to extract Terms (nouns and noun phrases);

b) using frequency information to find a subset of these Terms that are statistically important in the document,

c) matching these Terms with Topics in the KB;

d) performing disambiguation for ambiguous Terms (polysemous words that could refer to more than one Topic), preferably using a metric of "proximity in the KB graph" to other Terms in the sentence/paragraph/document scope;

e) creating new Topics in the KB for important Terms that are not already in the KB; and

f) indexing documents in the database based upon these Topic labels.

2. Implementation Design of Linguistic Analysis

The system analyzes documents off-line to determine what the document is "about" and label it with the appropriate meta-data so that Structuring can access it efficiently. The boundary between what processing is done in Linguistics and what is done in Structuring may be determined as necessary for implementation efficiency. The main idea is to do Linguistic processing to extract all the information needed from documents into meta-info records in the database. Then the Structuring processes can take sets of meta-info records (representing documents) to build spaces (preferably in "real-time" when someone is using the system). The goal is to allow the Structuring processes to be free from dealing with the document itself: they just use information extracted by Linguistics.

In order for the Structuring algorithms to make use of Knowledge about Concepts in organizing the documents, each of the extracted entities (i.e. noun phrases) in the document must be connected with the corresponding Concept in the KB (the term Topic to refers to a Concept as a nodes in the information space as opposed to nodes in the KB). The KB provides the fixed vocabulary of Concepts that are the possible labels for "what a document is about"; it suggests the relevant basic-level terms to use as meaningful general categories to put documents in. The use of a fixed vocabulary is extremely valuable for indexing and retrieval systems, for it prevents arbitrary and inconsistent strings (which are difficult to use to find documents later) from being used (e.g. Library of Congress Subject Heading System) In addition, the KB specifies constraints on the generation of new Concepts (especially collocations) discussed in the document. Further it is desirable for Linguistics to allow Structuring to identify the Concepts that are being discussed independent of the particular words used, by removing the dependency upon word choice or morphological inflection of a word referring to a Concept.

In sum, Linguistics can be viewed as placing documents in categories (Concepts) or as putting labels (Concepts) on documents; these are the same thing. The description here assumes that Linguistics starts with plain text files as InfoObjects. A document gatherer is used to collect documents (e.g. retrieve them from WWW), parse their formats (e.g. PDF, HTML), and provide Linguistics with chunks of plain text (where each chunk is grouped according to what part of the document structure it is from). In addition, a full-text indexer is used to index documents that enter the system; this will provide a parallel route to those mentioned below for retrieving documents later. It is also preferable to use a document splitting mechanism to decompose documents into parts, for example according to a table of contents or other document structuring information. This decomposition may be done manually or automatically.

i) Tokenizing, Stemming, Part-of-Speech Tagging

Input: text stream

Output: series of weighted tokens, suffixes removed, tagged with a POS symbol

An IR model of token frequency and relative importance. The Brill POS tagger may be used to perform the tagging. Alternatively, the INSO tagger may also be used.

ii) Noun Phrase Parsing

Input: series of tagged tokens

Output: series of noun phrases

Using a simple English grammar to find possible NPs: a first pass groups together tokens into noun phrases (e.g. NP=noun*; NP=adj+noun; etc.); a second pass attempts to assign attachment of phrases to each other (e.g. assign direct/indirect objects, attach prepositional phrases). The KB contains argument role information for different verbs to help with this parsing.

The general goal is for Linguistic Analysis to use knowledge-guided parsing to determine the relationships between concepts discussed in the document.

iii) Lexical Unification With Knowledge Base

Input: series of noun phrases

Output: series of corresponding KB Conceptds

Here are the different cases that are encountered in the process of unification:

NPs that have a single mapping in the KB:

In this simple case, a DB query is made for the NP as a Term in the KB and assign the appropriate ConceptId.

polysemous NPs (map to several Concepts in the KB):

Attempt disambiguation via proximity search from candidate Concepts to other active Concepts appearing in the context surrounding the NP.

unknown common nouns (not found in KB):

Make a temporary Concept (indicated by a status field) for the purpose of at least being able to determine co-reference with the same terms in other documents. To prevent too many unknown words in the KB, strict frequency thresholds are used before executing this pass.

unknown proper nouns:

A Proper Noun recognizer may be used to identify unknown tokens. The recognizer should suggest whether the token refers to a Person, Place, Organization, or Product.

unknown noun phrases:

As with unknown common nouns, make a temporary Concept (only for frequently occurring NPs—further, restrict here to only noun/noun and adj/noun NPs initially). Next, attempt to unify the parts of the NP (its component words) with Concepts in the KB. Next, find relation path between these constituents; this determines the relations that are assigned between the new Concept and the constituent Concepts, so that the new Concept is "linked" in the KB.

Annotation Enhancing

Input: all of the above information

Output: series of weighted ConceptIds that are implied topics of the document.

In addition to the above methods for literally extracting Concepts discussed by the document do determine what the document is "about", it is preferred to apply other algorithms that work with this extracted information to generate further Concepts that are good labels for the document. Here are some simple algorithms that may be used:

abstraction and clustering:

Given Concepts that the document is about, the document can be considered to also be about more general Concepts (supertopics) that subsume the explicitly mentioned Concepts (e.g. a document about "spaniels" is also about "dogs"). Stated otherwise, the KB relation of subtopic can be used to generate lists of terms which serve as evidence that a document expresses a more general Concept.

Clustering uses the same idea, but searches outwards in the KB from Concepts in the document to find points of contact (i.e. common Concepts) which can be given higher weights (e.g. a document that discusses Fords, wheels, tires, radios, dealerships, etc. can be also labeled with the Concept car).

query expansion evidence models

The method of query expansion used by Verity Topic-style programs is to translate a query term into a set of other terms whose presence in a document constitutes weighted "evidence" that the document is about the initial query term.

co-reference and weighing:

Determining correct co-references in cases of anaphora is a difficult problem. One approach is to obtain accurate counts of how many times Concepts are referred to; and then can the correct reference in analyzing clause-level relations between Concepts The two commonly occurring cases are pronouns (e.g. "There is Bob. He is neat.") and definite determiners (e.g. "Netscape is neat. The company is growing."). This approach is to using a simple metric of "assign reference to closest previous compatible type", which has about a 50% efficacy. In addition, KB synonym information may be used to determine co-referents, and to ensure that one Concept corresponding to several different terms in the document is weighted appropriately to reflect all of their occurrences.

deep parsing and summarization:

Alternate embodiments may use simple summarization by the technique of excerpting chunks of text in positions of high relevance, and then pruning off satellite clauses from these chunks.

When handling these linguistic issues, the KB may be used as a world model (and goal knowledge representation scheme) for building representations of the documents' content (e.g. entity-relation models).

B. Knowledge Base

1. Overview of the Knowledge Base

The Knowledge Base is a semantic network representing Topics, conceptual Relations between Topics, and Terms (lexical items) used to refer to Topics. The KB:

a) provides many-to-many mapping from Terms (word forms) to Topics (word meanings), allowing for polysemy and synonymy;

b) provides an application programming interface that projects any of its types of Relations into a smaller set that the Structuring process can use. The types of Relations include: refinement/generalization (taxonomic and conceptual subsumption/entailment) and association (common co-occurrence but not subsumption);

c) expands to include new hierarchies: contains many overlapping Topic hierarchies from different sources (e.g. Wordnet, Yahoo). Each Topic in each hierarchy is unified through a partially automatic and partially human process with conceptually matching Topics (evidenced by similar Terms and Relations) in the other hierarchies.

d) expands to include new Topics Terms and Relations automatically found in documents:

i) new Topics are made for important unknown Terms;

ii) association Relations are made for Topics which occur together in many documents;

iii) refinement Relations are made based upon linguistic criteria (e.g. the rule "a noun phrase is a taxonomic refinement of its head noun and a subsumption refinement of its modifying nouns" applied to the new Term "dog food" produces the Relations: "dog food" is a type of "food" [taxonomic refinement] and "dog food" is subsumed by "dog" [subsumption refinement]). This allows for the partially automatic large-scale extension of the KB to include a vast vocabulary and statistical information about which Topics are associated with each other.

e) includes a KB Management tool that provides a GUI interface for a human annotator to verify automatic additions and otherwise edit the KB.

2. Implementation Design of the KB

The knowledge base (KB) helps with three main areas of the system:

Linguistic analysis: POS tagging, parsing, and concept disambiguation

Structuring input: user's query expansion

Structuring process: finding/constraining/creating relations between Concepts

The KB is essentially a large collection of Concepts. For each Concept, there is stored the following information: a definition; a set of terms (and morphological information about these terms) that can be used to express it; a set of relations between it and other Concepts. In AI, this type of KB would typically be referred to as a "semantic network" with bindings to a "lexicon".

The KB is implemented as a set of tables in and access routines for a relational database.

A Knowledge Base Management Tool which allows for simple graphical is browsing and editing of all of the fields in the KB.

i) Relations Between Concepts

In principle, every relation between Concepts is reciprocal: for example, the relation "X produces Y" is reciprocated by the relation "Y is produced by X". (The exception to this rule is Attribute relations, which map from Concepts to literals.) It is desirable eliminate the redundancies of storing both of these separately, and simply store one version of the relation to represent both directions.

Each relation between Concepts is itself a Concept in the KB. For example, the relation "produces" (e.g. "Netscape produces Netscape Navigator") is a Concept in the KB, with terms and other relations. For example, "publishing is a type of producing" (note that hyponym (type-of)relations are somewhat different for nouns and verbs; for verbs, a better literal translation is "publishing is a manner of producing").

For Concepts that are also relations, the terms indicate what the appropriate inverse relation construction is. This also allows the system to represent arbitrarily specific relations in the KB, so that information is not lost when knowledge is entered into the system.

However, in one embodiment the Structuring Process and client only makes use of the most general basic types of relations between Concepts. In order to support this, there is also maintained a separate table (a DB RelType table) which stores the basic relations so that they can be readily accessed. Further, all instantiated specific relations that are entered into the KB must inherit from one of these basic relations; in this way, all relations that are entered into the KB can be of use to a module that only wants to deal with basic relations. Constructing the KB in this manner provide the infrastructure to accommodate a greater use of more detailed relational information as the Structuring and client become more sophisticated. For example, detailed relational information can be used by the client to further specify the appropriate layout and appearance of Concepts, or by Structuring to filter out different relation types to accommodate different user perspectives.

The basic relations are currently subclasses of two main types:

- "X is related to Y" (inverse same): more specific types of this relation include: sibling relations (from common parents or common children); association relations where no modifier is carried over (e.g. "Netscape produces Internet Software"); statistical unlabelled co-occurrence relations (i.e. parents of intersections, or statistical associations).
- "X is subtopic of Y" (inverse "is supertopic of"): this is a general parent/child relation, where the important defining factor is that more information is specified in the child than in the parent (e.g. a distinguishing feature, another modifying term, etc.). The more specific types of this relation include: taxonomic subtype relations; association related subtopic where a modifier is carried over (e.g. "Netscape produces Netscape Navigator"); meronymic part-of relations; statistical unlabelled dependence relations (i.e. intersections).

These are the distinctions between relations that are used in one embodiment of the Structuring Process and client; not all of these are used in the KB. The other commonly found relations in the KB (useful for hand annotation) are organized in an ontology within the KB itself, and can be examined there.

ii) Sources of Knowledge

The content of the KB may comes from several sources.

Wordnet

Wordnet is a freely available lexicographic research tool distributed by Princeton University. It consists of a large lexicon (roughly 100K terms) of concepts (called synonym sets in WN, for they are defined by the collection of words that express them, as in any thesaurus), and taxonomic relations between these concepts. Wordnet's coverage of the English language in general is very broad. The main organizing relation is hyponymy (the "is-a" relation, which is here called "subtype"); in the true spirit of taxonomy, every concept is considered to be a specialization of a more general concept (child=parent+distinguishing feature), and inserted into the hierarchy accordingly.

In one embodiment, the concepts in Wordnet serve as the foundation of the KB. In other embodiments, other lexicographic resources, such as Roget's Online Thesaurus and online encyclopedias, are integrated with Wordnet as well.

Mikrokosmos Ontology

The Mikrokosmos Ontology is a freely available resource, developed at CRL as the knowledge component of a knowledge-intensive machine translation project. It consists of small-medium (roughly 6K concepts) collection of general world knowledge concepts (with a specialization in the domain of corporate mergers and acquisitions) and the relations between them.

One embodiment uses Mikrokosmos because of the richness of explicit association relations it provides between world knowledge concepts.

Online Categories

Web sites which categorize information through hierarchical subject-based collections of pages are becoming increasingly common. The larger of these (e.g. Yahoo, Galaxy, etc.) effectively constitute "ontologies" wherein each page represents a subject (which maps to "Concept"), provides information about this subject (such as a collection of documents about the subject, and perhaps a definition), and relates the subject to more general (abstract) and specific subjects.

To incorporate online categories, it is preferable to use a crawler to download all of the pages from the site of interest. The desired information (e.g. ontological information) is extracted from the HTML, and saved in a knowledge base object. These objects are unified with the Concepts in the KB.

In one embodiment it is preferable to use Online Categories because they offer alternative dimensions of abstraction to standard taxonomic relations, and because they provide information about "real-world" concepts (e.g. companies) and domain specific concepts which Wordnet and Mikrokosmos do not cover.

Hand Knowledge Annotation

There are four main areas in which Hand Knowledge Annotation is desirable:

- to review and correct the automatic KB building/unifying processes;
- to add labels to unlabelled relations (e.g. from Online Categories);
- to create particular chunks of a KB for a customer:
- to enter an area of specific domain knowledge (e.g. from an expert) that is not recorded anywhere in a form that can be input automatically.

Partially Automated Generation

For customers that need specialized knowledge related to their domains, it is useful to develop a few simple utilities to case the acquisition of this knowledge: a utility that finds commonly occurring unknown noun phrases in a typical sample of the customer's documents, and puts these in a buffer for a human to add to the KB in the right place; a utility that takes lists of terms under headings in a simple file format and inserts these into the KB awaiting further human specification.

Unifying Knowledge Sources

In a preferred embodiment, the system combines the information from these different sources to form one unified KB. For example, the KB should respond to a request for all of the children of the Concept "computer", by providing the union of Wordnet's, Yahoo's, etc. subtopics. It is preferred not to have multiple entries for the same Concept or relation between Concepts. Overall, the knowledge in each KB is made much more valuable if it can be linked to the other KBs.

In order to perform the unification from these different sources, one embodiment uses an algorithm that attempts to automatically determine which Concepts in source A map to

which Concepts in source B. The algorithm uses as evidence for a match the gradated results of comparisons between the names, definitions, and lists of related Concepts between A and B. After initial automated unification, the KB is reviewed by a human operator to provide the final selection and unification of concepts from the disparate knowledge sources.

C. Structuring Process

1. Overview of the Structuring Process

The Structuring Process (also called space-building) generally means:

Given a set of documents (or a query from which to select a set of documents), create a Space (a graph) of Concepts that permits navigation of the set of documents. Each Concept node in the space indicates what information is available about that Concept (i.e. what the system understands about the information) through the presentation of related topics in space; this effectively defines the local "vocabulary" the user has for indicating the next step in the "dialog".

Structuring is the system's ability to organize information. The information spaces (or simply "spaces") resulting from the Structuring Process assign coherent and consistent meaning to the relative positions of Concepts in the space. Unlike fixed category schemes, the resulting Spaces are dynamically constructed to reflect the Concepts discussed by a specified document set; only Concepts discussed in the available documents are shown in the Space. Automatic categorization of documents in a Space enables visualization of the Concepts.

The main process of Structuring involves recursively finding common Concepts that can group documents to provide coverage over a document set, and finding subtopics of these that provide distinction between these documents to yield smaller document sets. The KB provides constraints on what Concepts can be used to group other Concepts, and what Concepts can be used to subdivide other Concepts. In order to provide good organizations of document sets, knowledge is required; the structure of the space must be consistent with general world knowledge.

In one embodiment, the Structuring Process builds a Topic graph for a set of documents in response to a user query. The Structuring Process includes:

a) using the Topic labels derived from Linguistic Analysis to retrieve relevant documents and initial Topics;

b) using the KB to expand the query and constrain the choices of Relations between Topics in building the graph; and

c) selecting the set of Topics that most efficiently organizes the documents by recursively creating sub-Topics that provide coverage over and distinction between each Topic's documents.

The benefits of this system and its resulting Topic graph include:

1. The graph contains only those Topics relevant to the document set and user query, and useful for dividing up the documents efficiently.
2. Topics in the graph are consistently semantically related, allowing a user to follow a "train of thought" through connected Topics.
3. The query can be small (a word or two): the graph will include all documents possibly relevant to the query, allowing the user to refine the query by selecting Topics that increase the precision.
4. The graph is a representation of the document set, effectively providing a "summary" at each Topic level of the main sub-Topics discussed in the documents associated with that Topic.
5. The graph includes the relevant and useful sub-graphs from any number of existing hierarchies which are combined in the KB, allowing for broad coverage.
6. The graph includes new Topics generated to further refine Topics for which there are no refinements the KB (see below).

The three major operations performed by Structuring are:

i) Finding Documents That Match a User's Query

This involves searching the document DB after a user has typed a simple initial query to indicates the area of interest. The set of documents can also be constrained with a more traditional "select" on standard relational fields (e.g. date, author, source).

Input: query

Output: set of documents

The KB helps by broadening the query to extend the set of documents that are matched (recall), as well as pruning the possible interpretations of the query to avoid false positives (precision). For example: if the query consists of more than one term, the KB can aid in parsing it, as in Linguistics, the pruning can also occur by similar disambiguation to that done in Linguistics. Each query term is expanded to include morphological variants, synonyms (standard thesaurus information), subtypes ("animals" is extended to "dogs"), subparts ("dogs" is extended to "paws"), subtopics of the query term ("computer" is extended to "computer software"), Concepts related to query term ("netscape" is extended to "internet"), etc. In other words, all the KB Concepts which are related to or subsumed by the query term are also included in the search, so that it needs not rely on matching an exact word, but can instead match the general concept of interest.

ii) Organizing the Results in a Structured Space

This involves actually building the space that represents the document set. This is the main function of Structuring.

Input: set of documents' meta-data (sets of doc-Concepts); optional query-Concepts

Output: a Space

The Structuring process builds the space by finding the smallest set of Concepts that can categorize all of the documents that match the query, and that represents the content of these documents (i.e. the Concepts and Relations they discuss). The Knowledge Base provides information about how these Concepts can be organized in the Space according to their semantic relations, to produce a "reasonable" arrangement of topics (which documents go together in which categories, what Concept names are coherent for categories in a context, which Concepts are conceptually contained by other Concepts, which Concepts are generalizations of other Concepts; in general, how two Concepts semantically relate and should be placed relative to each other in the Client layout, etc.). This knowledge is combined with the results of Linguistic analysis to represent how documents and Concepts are related.

The output of Structuring is a Space with a structure that serves both to organize the documents that match the query into smaller coherent sets (to allow for progressive refinement of the user's search for particular information), and to inform the user about Concepts and Relations between them.

The problem, therefore, is one of automatic categorization of documents: putting documents in the right categories, and putting subcategories in the right categories.

The Structuring algorithms require Knowledge to give the user a coherent set of choices at each stage in the dialog with

the system, as the user articulates their request (through their gestures of moving towards the Concept of interest) with greater specificity. Movement in the Space accumulates a context that constrains what can be shown at deeper levels; the choices a user has when deep in the space reflect the choices the user has made to get to this point (as would be the case in any "dialog").

The general criteria for a Space are as follows:

- similar documents and subtopics are clustered together: topics group similar things
- topics should allow for progressive disclosure of information from general to specific
- topics should have links to all related information (related topics) to the topics, with each of these related topics being positioned consistently in the space (in terms of the type of relation to the current topic)
- topics should be at the proper level of generalization to indicate what the document set discusses: the set of top level topics serves as a summary of topics addressed by the document set
- the resulting space should be well balanced, and should use coherent and useful terms (such as basic level terms; avoid terms like "bureau"); the space should have minimal depth, given a maximum branching factor
- minimize the degree to which topics imply that the documents discuss things that they do not actually discuss
- topics should provide coverage over the documents and distinction between documents based on the choice of topics; the Space should have nodes to distinguish between arbitrary document set sizes (e.g. a user can specify a threshold: they always want to have distinguishing subtopics available whenever there are more than X documents about a topic)

The query-Concepts reflect constraints on what Concepts should appear on the top of the Space; subtopics are chosen for the rest of the Space that relate to these query-Concepts. The query-Concepts are initially simply the Concept-unified words that the user specifies in an input text query; however, the construct is generalizable to take into account other top-level constraints on the space, such as user profile information (a set of Concepts that a user always wants to see at the top of the Space), customer specific layouts (an "upper Space" that fixes certain Concepts in certain positions and allows the algorithms to "flesh out" the Space beneath these), etc.

The doc-Concepts are the set of Concepts from the document meta-data; these are the Linguistics labels of what each document is about. Structuring can be seen as the recursive process of finding those nodes that are the intersection of the nodes related to the current Concept (initially the query-Concepts) and those related to the doc-Concepts.

a) Algorithms

Abstractly, the main algorithm for Structuring consists of selecting the relevant subgraph of the KB to hold the documents, and collapsing (eliminating) unnecessary intermediary links between nodes. This involves two phases: 1) search upwards from each doc-Concept to more general Concepts to find common parents that can be reified as nodes in the Space; 2) search downwards and outwards from query-Concepts to more specific Concepts to find nodes that efficiently divide the set of documents into categories; in-between Concepts that are not branching points are removed (e.g. not put in a Space as a Topic). The Space is a tangled web of interconnected links (not a strict hierarchy); this allows "chunks" of the KB to appear (i.e. be accessible from) many different places. In addition to this Knowledge based Structuring on Linguistic meta-data, the central Structuring algorithms may be extended to take advantage of extra-linguistic information, such as hyperlinks in the documents.

In the top-down part of building a Space, there are some Concepts that have no further subtopics in the KB; this requires that the system "generate" new Concepts to further subdivide the documents related to the current Concept (if necessary to accommodate user's threshold. (The depth of the Space can be dynamic as appropriate for the document set). The goal is to have as many different nodes as necessary to distinguish between the documents and as few as necessary for covering the topics discussed by the documents. In these cases, Structuring examines the set of Concepts which the bottom-up algorithms have selected (which divvy up the documents) and try to find "paths" in the KB between these and the current Concept. (This is effectively trying to find relation labels for the co-occurrence associations between the current Concept and the bottom-up Concepts.) Structuring computes the cost of a path through KB according to weights on relation types; this allows Structuring to avoid semantic anomalies (e.g. "cat dog") while producing coherent nodes (e.g. "pet dog"). Structuring then generates new nodes for the set of shortest paths that provide the desired coverage and distinction over the documents about the current Concept. For each of these nodes, Structuring uses a label-generation routine to indicate how to correctly combine the constituents into a well-formed term (e.g. "A produces B" generates "A[possessive]B").

Upon completion of full phrase-level Linguistics parsing, Structuring will be able to use the internal structure of the parsed NPs to determine the nature of the relations between the constituents. The relation paths will allow Structuring to find and distinguish between otherwise confusing ambiguous intersections. For example, Structuring would be able to identify whether "music" and "computers" are co-occurring in the context of "computer music" or in the context of "people who like both computers and music"; Structuring would also be able to generate the right Concept label that puts the modifier in the right place. In the absence of this, Structuring can still use unlabelled information from a statistical analysis on a portion of the space that is not well mapped out by knowledge, thereby taking advantage of statistical processing to handle new things that are outside of the fixed domain of the knowledge base. The statistical approach starts with all the co-occurring frequent words found in the documents and uses the statistically independent ones as the more general topics in the space. The resulting space has top-level terms that are the starting points of trees where the nodesterms at each level indicate that documents beneath that node contain intersections of the current term and the previous terms up the tree. Overall, combining knowledge with statistics can be useful in cases where there is a consistent knowledge domain but a variation of numerical or other isolatable values within the organization. The knowledge part is important for organization and presentation, the statistical information is important to filter information before it gets to the organization stage.

iii) Updating the Space

There are several cases in which Structuring will need to compute an update to an existing space:

- When new documents are added to the database, these need to be placed appropriately in the Space. If many have been added, Structuring needs to recompute the

Space so that its organization reflects the new availability of documents.

The user can simply enter another text query term at any point and the system will create an intersection of this and their current results, using the path finding algorithms to attempt to relate the current Concepts to the new search term.

The user may want to indicate that a portion of the Space should remain the same, while another portion is recomputed.

2. Implementation Design of the Structuring Process

In a preferred embodiment, the Structuring Process includes the following steps:

1. The user query indicates what documents to build the graph around:
   a) The query can include Filters, such as a simple relational database select statement upon any field of the document record (source, author, date, full text index).
   b) The query can include Terms for Topics in the KB. The system automatically expands the query to include all Topics that have KB association or refinement Relations to the query Topics. For example, the term "dog" is expanded to include "leash," "spaniel," and the like. The expanded set of Topics is matched against the Topic labels that documents are indexed on.
   c) The query can comprise several Terms and Filters which specify top-level nodes in the graph. The rest of the automatically created Topics will attach to this top level.
   d) The query is executed to retrieve the matching documents and their Topic labels.
2. A bottom-up phase selects all the possibly relevant Topics from the KB:
   a) For each of the document Topic labels, all of the Topics connected by association and generalization Relations in the KB are retrieved, recursively.
   b) This effectively selects a sub-graph of the KB that includes all known abstractions of the document Topic labels.
3. A top level set of Topics is selected to provide coverage over and distinction between the matching documents:
   a) If the user query included Terms for Topics, use these Topics as the top level.
   b) Otherwise, choose the top level set of Topics from the most abstract Topics in the KB sub-graph.
4. A top-down phase recursively creates sub-Topics of these selected Topics, dividing the documents into smaller sets:
   a) For each Topic, evaluate whether the sub-Topics connected to it via refinement Relations cover a smaller set of documents.
   b) Choose the smallest set of sub-Topics that provides coverage and distinction.
   c) Eliminate Topics that are not part of this set (e.g. alternate refinements of the Topic from the KB subgraph).
   d) Compress the chains of Relations between Topics as necessary. For example, if the KB has refinement Relations between the Topics "animal" and "dog", and then between "dog" and "spaniel", but all of the documents about "dog" are also about "spaniel" then the "dog" Topic is removed and "spaniel" is connected directly to "animal". This ensures that the Topic labels are always as specific and relevant as possible.
   e) Recursively proceed until each Topic only has a specified maximum number of documents attached to it, and needs to be divided no further.
5. New sub-Topics are created to divide the documents associated with Topics for which the KB has no refinement Relations to possible sub-Topics.
   a) New sub-Topics are created as intersections of existing Topics: other Topics that have partially overlapping attached document sets are combined with the "unrefinable" Topic.
      For example, if there are no refinements of the Topic "spaniel" in the KB, but there are more than a specified maximum number of documents attached to it, and if the document sets attached to other Topics such as "shedding", "adoption", etc. are partially overlapping with the document set attached to "spaniel", then sub-Topics of "spaniel AND shedding", "spaniel AND adoption", etc. are created, such that the intersection is a refinement sub-Topic of each of its constituents, designating a more specific subject with fewer documents attached to it.
   b) Topics that have associated Relations in the KB are preferred for the creation of intersection sub-Topics, such that the resulting sub-Topic is a semantically related compound to the Topic. In cases where there are also no useful associated Topics, any Topic can be used.
   c) Intersection sub-Topics are themselves recursively further refined as needed by refining either of their constituent Topics, or by intersecting with another Topic.

Recursively proceed until each Topic only has a specified maximum number of documents attached to it, and needs to be divided no further. Intersection sub-Topics are themselves recursively further refined as needed by refining either of their constituent Topics, or by intersecting with another Topic.

D. Information Spaces

Spaces are now interconnected graphs (overlapping hierarchies) of Topics connected by TopicLinks, starting at Static Upper Level roots. Each TopicLink relates two Topics using a pre-defined RelationType. Each Topic has Labels, consists of several Focus Elements, and can have Multiple Parents.

1. Topics

Each Topic effectively serves as an expression of a "query" or "select statement" on a set of documents. The Topic is defined as one or more Elements which each restrict the set of documents that are "contained by" that Topic; the Topic contains that set of documents that is the intersection of the sets contained by its Elements.

Each primitive Element is either a Concept from the Knowledge Base (e.g. dog) which selects InfoObjects that Linguistic analysis has labelled as being "about" that Concept (content meta-data), or a Filter (e.g. source=Mac Week) that selects a set of InfoObjects based on their file meta-data (e.g. date, author, source, keywords assigned by author).

For example, a Topic could be "documents from June Macweek about computer chips and democracy" where "June" and "Macweek" are Filter Elements and "computer chips" and "democracy" are Concept Elements.

In one embodiment, it is desirable to have each possible Filter itself be a Concept so that these types of Elements may be unified. In the rest of this document, "Concept" is used to represent all types of "Elements".

In some embodiments, Topics which are "unions" of Concepts may be used instead of "intersections", as when a user asks for documents from June and July.

TopicLinks specify a change in a Concept in a Topic. Movement between Topics (via TopicLinks) in the Space graph entails either adding an Concept, removing an Concept, refining an Concept, generalizing an Concept, or changing (swapping) one or more Concepts. These break down into three categories of "narrowing" (subtopics), "broadening" (supertopics), and "jumping to some other place" (associated topics). Each of the types of movement has distinct semantic properties, and is represented by a different RelationType for the TopicLinks.

On the level of the Conceptual Model for the Space Structure, these movements define the discrete gestures that the user can make in the discourse with the system. Therefore, movement in the graph entails modifying an evolving "context stack" of Concepts.

2. RelationTypes

The system architecture is set up to accommodate an unlimited variety of RelationTypes. A mapping from these RelationTypes to a smaller pre-defined set that the client is prepared to meaningfully display is preferred to enable reuse of a client with new RelationTypes.

One set of RelationTypes includes 8 relations, divided into the 3 categories of sub, super, and associated topics.

The RelationType between a source Topic and a destination Topic reflects the semantic relation between these Topics, and the relationship between the document sets associated with each of them.

Not all RelationTypes will be available at each Topic.

i) Subtopics

A subtopic represents a refinement of the document set, a specialization of the source Topic "query". Therefore, subtopics match the "container" metaphor of the client: each subtopic's document set is "contained by" the source Topic's document set; the destination Topic is conceptually contained by the source Topic.

The basic idea is that if a document is "about" a subtopic, e.g. jazz, that entails that it is also loosely "about" the supertopic music. The three different kinds of subtopics listed here reflect different ways in which a user could think of refining the topic of interest. The types of subtopics include:

subtype subtopics: RLTN_SUB_TYPE: "sub-type"
music ->jazz
software ->business software
dogs AND music ->dogs ANDjazz

These are examples of taxonomic refinement of an Concept along an IS-A dimension.

related subtopics: RLTN_SUB_RELATED: "sub-related"
music ->music clubs
car ->engine
business ->business software
dogs AND music ->dogs AND music clubs

These are examples refinement of an Concept area of interest to a more specific Concept that is not a subtype. This may be used to define part-of or member-of relations, and typically involves using the current Concept as a modifier of another Concept (although in some cases, the modifier is implicit, e.g. car ->[car] engine).

(Note that the resulting Concept is still one unified Concept in the Knowledge Base that has a particular meaning (though it combines other Concepts); as opposed to intersection subtopics, which can combine two separate Concepts but do not form a single unified Concept.)

intersection subtopics: RLTN_SUB_INTERSECTION: "sub-intersection"
music ->music AND clubs
UNIX->UNIX AND database
dogs AND music ->dogs AND music AND clubs

These are examples of addition of a further Concept to the Topic.

This is how the "filtering" examples is preferably handled: given a set of documents about a certain topic, a new topic (e.g. a filter for database products) is added as an "intersection" to further refine the document set. The new topic can be of unknown semantic/conceptual relation to the current topic.

ii) Supertopics

A supertopic represents a broadening of the document set, a generalization of the source Topic "query"; these are simply reciprocals of the subtopic relationships. The types of supertopics include:

supertypes: RLTN_SUPER_TYPE: "super-type"
jazz ->music
music clubs ->clubs
dogs AND music clubs ->dogs AND clubs

related supertopics: RLTN_SUPER_RELATED: "super-related"
music clubs ->music
dogs AND music clubs ->dogs AND music

un-intersections: RLTN_SUPER_INTERSECTION: "super-intersection"
music AND clubs ->music
music AND clubs ->clubs

iii) Associated Topics

An associated topic or related topic is a Topic that is generally related to the source Topic, but not in a strict generalizationspecialization sense. Conceptually, it is "to the side" of the source Topic. These RelationTypes may be understood as "See Also", or "Discovered Associations", etc. They will be "warps" to other parts of the Space graph, as opposed to subtopics and supertopics, which move along more consistent hierarchical dimensions.

In some cases, associated topics may be siblings (having common parents, or common children), but not always. (In some case associated topics may have common parents in the Topic graph, because they will represent some overlapping documents, but they may not necessarily have common parents in the Knowledge Base Concept graph.)

The types of associated or related topics include:

discovered associated topics: RLTN_ASSOC_DOC: "assoc-doc"
Microsoft ->Explorer
dogs AND Microsoft ->dogs AND Explorer

This is a swap of Concepts that are strongly associated in the document set for the Space. These serve as a representation of the "content" of the documents, of what the document set "says". Initially, these may be determined by statistical co-occurrence; alternatively, the may be the result of deeper parsing towards "text understanding", with the goal of being able to represent the "way" in which the Topics are associated (e.g. "Microsoft produces Explorer"). This deep level of analysis is similar to semantic summarization.

knowledge base associated topics: RLTN_ASSOC_KB: "assoc-kb"
markets ->products
dogs AND markets ->dogs AND products

This is a change of a Concept with a Concept that the Knowledge Base considers to be strongly associated. This is the mechanism through which system developers can add specific knowledge to link associated Topics.

3. Focus

A Topic can be composed of many Concepts, and any one of these may be the "subject" of the relation to a destination

Topic. The subject of the relation the "Focus". Among other things, this allows the system to indicate which of several intersecting Concepts in a Topic are to be further refined.

For example, from "computer companies AND jazz", there are following different subtopics:

->computer companies AND bebop (reltype=subtype subtopic, focus=jazz)

->Apple AND jazz (reltype=subtype subtopic, focus=computer companies)

In some cases, it might not be meaningful to say what is the focus:

->computer companies AND music AND food (reltype=intersection subtopic, focus=[any])

The thrust of Focus is around supporting changes of Focus between different Concepts within one Topic, as in the above examples. In addition, it is desirable to implement changes of Focus within one Concept to indicate what part of the Concept is being modifiedrefined, as in the following:

->portable computer companies AND jazz (reltype=subtype subtopic, focus=computer)

->California computer companies AND jazz (reltype=subtype subtopic, focus=companies)

This can be difficult, because many semantic anomalies are possible when "mixing and matching" terms within a topic; e.g., refining on pan in pan flute would be inadvisable.

4. Multiple Parents

Every Topic can now have multiple parents, ie. supertopics.

These can be partially distinguished based upon Focus and RelationType.

For example, from "computer companies AND jazz":

->jazz (reltype=un-intersection supertopic, focus=jazz)

->computer companies (reltype=un-intersection supertopic, focus=computer companies)

->computer AND jazz (reltype=related supertopic, focus=computer companies)

->companies AND jazz (reltype=supertype supertopic, focus=computer companies)

->computer companies AND music (reltype=supertype supertopic, focus=jazz).

Appendix B

A Scaleable Camera Model for the Navigation and Display of Hierarchical Information Structures

1. The Problem

Hierarchies are powerful organizations for capturing underlying order among data objects, but are difficult to display such that humans can easily understand their overall organizations and traverse them in order to locate desired data objects. (It should be noted that the term "hierarchy" here loosely includes the more general "graph" structure of which the hierarchy is a special type. This looseness is useful because even when a context applies to the general case, the hierarchy will still be the most important case. When the distinction is important, the discussion will make it clear.)

A. Traditional Hierarchical Displays

1. Outline Mode

Microsoft's "Window's Explorer" is an example of an attempt to display such a is hierarchy (in this case, a hierarchical file system), and allow users to traverse and manipulate it. Most word processors offer an outline mode view which is a powerful feature allowing writers to visualize and work within a hierarchical document. Outline mode works fine for small hierarchies but becomes difficult and tedious for large ones. The problem is even more difficult for other non-hierarchical graph structures.

2. Absolute Coordinates

Another approach is to model and display hierarchical containment as self-similar branches of ever smaller scale within two or more dimensions. Display and navigation techniques then allow users to magnify and expand regions of interest to both visualize overall structure and to find data objects of interest. This sort of multi-resolution visualization can be seen as either the ability to manipulate and scale the data structure, or as the ability to translate and scale one's point of view. Both are equally valid and equivalent ways of looking at the same operations.

Pad+ is an example of a system using this technique in two dimensions. Graphical data can be scaled to any size and placed anywhere within a continuous 2D space. The user can expand, contract, and pan around to view any part of the space at any scale. Pad+ was not designed specifically to support hierarchical information structures, but it does not preclude it.

i) Problems With Absolute Coordinates

a) Modeling

Perhaps the biggest problem with modeling hierarchies in absolute coordinates is the difficulty of maintenance. For any object to be placed in a hierarchy, it must be assigned coordinates placing it rigidly within that structure. It is the equivalent to modeling a house brick by brick. It is not enough to simply state where the house is to be placed, one must determine where every brick goes. This rigidity also makes it difficult to alter hierarchies once they have been created. So as with placing a house, moving a house entails providing new coordinates for each brick.

b) Numerical Instabilities

Even though it is mathematically simple to think of a system that allows absolute coordinates to be computed within a hierarchy, in order to be used on a typical computer, those coordinate values must be stored with some finite precision. As the number of levels in a hierarchy grow beyond even a moderate number, even double precision numbers quickly run out of resolution to adequately represent numbers at very different scales. True, there are ways to store numbers with arbitrary precision within computers, but the cost in additional memory and especially in processing time is usually prohibitive.

3. Relative Coordinates

Relative coordinate modeling means describing each object in its own coordinate system and then using or instancing them wherever they are needed in the overall structure. At display time, each instance of an object takes its scale, position and possibly other attributes from its parent. This is a very powerful and common technique in computer graphics. Its greatest power is in its flexibility since moving or reparenting a node simply involves moving or transforming a link. The results of such changes are only observed at display time.

The use of relative coordinates also helps greatly with the issue of numerical precision. Each object can be modeled using the full range of numerical precision. It is only when extremely large and small objects must be rendered in the same coordinate system that numerical problems creep back in. Usually, there is plenty of precision available in traditional single and double precision numbers to have the added luxury of modeling each object in terms of the most natural units for those objects: feet or meters for human scale objects, Angstroms for molecules, etc.

Of course most problems resulting from the choice of coordinate systems can be worked around, but there is value in choosing the right solution for each need.

B. Non-Hierarchical Structures

So far this document has only discussed strict hierarchical models (i.e., true trees). It is often very important to be able to represent, display and navigate more general graph structures which may contain loops. Such structures would be impossible to represent completely in absolute coordinates. To see this, imagine a node A that contains node B. Being a sub-node, node B (or a copy of it) would be need to be modeled smaller and contained within node A when using an absolute coordinate system. If node B also contained node A, then it would need to contain a smaller copy of A that would contain an even smaller copy of B, ad infinitum. In a system which displays only a portion of a general graph at a time as a user navigates from node to node, a relative coordinate system is the best choice since copies of these nodes at varying scales would never need to be produced. The camera could descend from A into B and then deeper again back into A as many times as desired.

II. Architecture of Solution

The key insight to this solution is in the combination of three techniques:

1. A camera restricted to always face along a fixed vector but free to pan and zoom,
2. The use of a relative coordinate system of bounded sub-spaces, each one completely containing its children (with the assumption that the camera transitions between spaces by entering or leaving these bounded spaces), and
3. A camera volume which shrinks (or spaces which expand) in proportion to the camera's depth within its current space.

Of these three techniques, perhaps only the last one (the scaleable camera) is new, but the combination of all three is what is particularly useful, novel and non-obvious.

Restricting the view direction vector largely helps users from becoming "lost", and provides a uniformity of display that helps guarantee that the display makes sense at all times. In the current implementation, a three-dimensional space is used in which deeper levels of a hierarchy always appear further behind the display surface than shallower levels. A perspective effect is achieved by scaling objects smaller in proportion to their depth in the current display, The navigation technique models the user as being at a certain position and of a certain size within the display, so objects appear smaller the further they are in front of the user, and objects behind the user are not drawn. The current implementation uses a parallel projection, though this technique should work equally well using perspective or other projections.

The use of relative coordinate systems is important for several reasons:

- It avoids numerical instabilities that would otherwise make it difficult to model hierarchies of arbitrary depth.
- It makes it possible to model an arbitrary node independently of any ancestors.
- It makes it possible to modify any part of a hierarchy without having to recursively apply transformations throughout that part of the hierarchy.

In addition to the use of relative coordinate systems is a containment model important to display and navigation. Although each node carries with it its own coordinate system, a useful restriction is the assumption that everything it contains will not extend beyond a certain range—in this implementation, a unit box. This restriction is important for two reasons:

- It allows for the display-time culling of complex regions of the model which are outside the current viewing volume, and
- It clearly delineates a spatial domain for each node so even though the user moves through a continues information space, it is always clear where they are in terms of node traversal. That is, the user is always clearly physically in the scope of exactly one node.

This architecture also supports the treatment of non-hierarchical graph structures which is important because not all useful organizations of data can be captured in strict hierarchies. It is the use of relative coordinate systems that makes this possible since it becomes simple and natural make any node appear to contain any other related node even where the semantic relationship is not one of strict containment. As described in the previous section, two related nodes may each contain the other so that the resulting display gives a "hall-of-mirrors" or "infinite recursion" effect that would be very difficult to achieve with a model based on an absolute coordinate system.

III. Method

The essential components of the system are the nodes (topics) and the camera. Each node represents a complete and bounded information space consisting of information directly relating to that node (title and other graphical annotation, plus data objects), plus a set of related sub-spaces (other nodes). The camera represents the user's point of view that is always located in the space of one particular node (called the current node or topic), and which can move from node to node by entering related sub-spaces or backing up into previously visited spaces.

Although there can be many different ways that one node can be related to another, these relations can be categorized into two basic types: Those relations that represent complete containment—such as part-of or is-a relations—called "subtopics", and those that do not—called "related" topics. This distinction is very useful in that the display can use knowledge of this distinction to infer and display the user's context within a space. When the user moves from one topic to a subtopic, they are narrowing the scope of their context, which can be displayed as a refinement stack of topics. Traveling to a related topic, on the other hand, takes the user to a completely new space and generates a fresh refinement stack. Another interesting type of related topic is the "parent" relation. Any given topic can naturally be contained by several different topics simultaneously, and those parent topics can then be represented as a type of related topic within the given topic.

The user can drive the camera forward, back, left, right, up and down simultaneously, and the volume of space displayed at any time is centered on the position of the camera. An important feature of this method is that the size of the volume of space displayed is proportional to the distance from the camera to the back wall of the topic space in which the camera is currently navigating. Another important feature is that the speed with which the camera moves is also proportional to that distance. This way the user can drive continuously forward, yet never reach the back wall. The very act of moving forward (deeper) into the space therefore gives the effect of spatial refinement, while moving backwards becomes spatial generalization. Sub-spaces and other objects positioned against the back wall of a topic will appear to grow and shrink much like the effect of expanding and contracting in infinitely stretchable sheet. An important feature of this method is that objects can be placed anywhere within a space (i.e., not only against the back wall) making the model appear within an infinitely expandable three-dimensional space instead of simply on a two-dimensional sheet.

IV. Data Model

The following description only describes the current reduction-to-practice and only serves to describe one possible implementation of the Scalable Camera design.

The system components comprise:

A scene graph, also called a "space",

Graphical objects representing instances of nodes of that graph, and

A camera.

A. Scene Graph

A scene graph can represent any information organization to be viewed. Each node in that graph represents a topic. A related structure is constructed in the display system which consists of graphical instances of those topics. These graphical instances are used as the nodes of a 3D display hierarchy. They contain several important pieces of data:

1. A reference to the graph node that they represent (for access to title, summary and other features to display),
2. Size and positional data placing the node within the coordinate system of its parent, and
3. Zero or more child nodes.

A more flexible version which is currently not implemented would not have a graphical topic's size and positional data stored within that graphical topic, but would instead have that data simply be associated with it and stored within each parent that contains that graphical topic.

The fundamental operations on a graphical topic are:

1. The ability to arrange, or "layout" the children and other graphical elements of chat topic,
2. The ability to render those graphical elements onto a display given a display context defined by a viewing camera, and
3. The ability to select graphical elements from within the displayed region of the topic given a display context plus a selection point or region on a display.

B. Unit Box

The fundamental, low-level data structure on which the implementation is based is called the "unit box" which simply contains the position and size information mentioned above. In a more general 3D graphics system, this information would normally be captured in transformation matrices, but due to the restricted nature of the camera motion and the fact that graphical sub-spaces are never rotated, a much more compact and efficient representation can be used. Implicit in the relative coordinate systems that unit boxes represent are the bounds, or limits of these sub-spaces. Currently, the implicit bounds of a unit box range from negative one half to positive one half in the X (left to right) and Y (bottom to top) dimensions, and from zero to one in Z (coming out of the screen).

The fundamental operations performed on unit boxes are tests for containment of points and other boxes, plus computation of the unit box representing one unit box in the coordinate system of another.

C. Camera

The only fundamental data stored in a camera object in this system is its position (called the "eye point") within the unit box of some current graphical topic's unit box. This plus a description of a destination display surface defines the essential features of a display context.

The fundamental operations on camera objects are:

1. The ability to move the eye point (with constraints to restrict it within legal bounds),
2. The ability to map between relative modeling coordinates and screen coordinates, and
3. The ability to smoothly animate (interpolate) the eye point from one point to another.

Each time a camera moves forward within the space of a current topic, a test is made to see whether it has entered one of that topic's children's sub-spaces. Likewise, each time a camera backs up, a test is made to see whether it has backed out of a sub topic and returned to a parent topic. Either of these cases triggers a change of context. The position of the camera is then reset to be last position of the camera as expressed within the coordinate system of the new current topic, and navigation continues within that new context.

The volume of space displayed at any time is the cube with edge length equal to the distance from the camera to the back wall of the current topic's space, and centered half way from the camera's eye point to that back wall. A little extra space either horizontally or vertically may also be displayed beyond the bounds of that cube to account for non-square display surfaces. All child nodes within that volume are displayed, as are their descendants down to some maximum number of levels depending on desired display density and rendering time allowed. This automatic scaling of the camera volume proportional to its relative depth is one of the key features of the system.

We claim:

1. A system for displaying information, comprising:

a) an information structure having a plurality of semantic entities, each semantic entity having:

(1) a navigable link to at least one other semantic entity;

(2) a graphic object for representing the semantic entity on a display screen;

b) a display window having a variably resizable display area, and a selected information density; and

c) a display engine that displays graphics objects of a selected number of semantic entities, the semantic entities selected from the information structure in accordance with the selected information density.

2. The system of claim **1**, wherein each graphic object is capable of being displayed at any of a plurality of sizes.

3. The system of claim **2**, wherein the size of each displayed graphic object is determined in accordance with the selected information density of the display screen.

4. The system of claim **3** wherein the information density is a constant and the size of each graphic object is a function of at least the information density.

5. The system of claim **1** wherein the information density is a constant and the number of semantic entities selected is a unction of at least the information density.

6. A method for displaying semantic information in the form of graphic objects, comprising:

a) storing an information structure having a plurality of semantic entities, each semantic entity having:

(1) a navigable link to a plurality of other semantic entities; and

(2) a graphic object for representing the semantic entity on a display screen, each graphic object capable of being displayed at any of a plurality of sizes, and having a shape;

b) displaying a first graphic object of a first semantic entity;

c) displaying within the shape of the first graphic object the graphic objects of each semantic entity linked to the first semantic entity; and

d) dynamically scaling the size of the displayed graphic objects to maintain a selected information density of displayed data.

7. The method of claim **6** wherein the information density is a constant and the size of each graphic object is a function of at least the information density.

8. A method for displaying semantic information in the form of graphic objects in a display window, comprising:

a) storing an information structure having a plurality of semantic entities, each semantic entity having:

(1) a navigable link to a plurality of other semantic entities; and

(2) a graphic object for representing the semantic entity on a display screen,

wherein selected ones of semantic entities semantically contain at least one other semantic entity;

b) displaying in the display window first graphic objects of a plurality of first semantic entities from the information structure, the display window having a variably resizable display area and a selected information density;

c) displaying a cursor in the display window;

d) receiving a user input to move the cursor toward at least one of the displayed first graphic objects;

e) simulating movement toward a first displayed graphic object by:

(1) increasing the size of the displayed first graphic objects; and

(2) displaying second graphic objects of second semantic entities contained by the first semantic entities;

wherein the size of the displayed graphic objects is determined in accordance with the selected information density of the display window.

9. The method of claim 8 wherein the information density is a constant and the size of each graphic object is a function of at least the information density.

10. A method for displaying semantic information in the form of graphic objects in a display window, comprising:

a) storing an information structure having:

(1) a plurality of levels of semantic containment, each level of semantic containment having:

(a) a plurality of semantic entities, each semantic entity having:

i) a navigable link to a plurality of other semantic entities;

ii) a graphic object for representing the semantic entity on a display screen;

such that each semantic entity either semantically contains at least one other semantic entity, is or semantically contained by at least one other semantic entity;

b) displaying in the display window graphic objects of at least one semantic entity from an Nth level from the information structure, the display window having a variably resizable display area and a selected information density;

c) for each semantic entity from the Nth level that is displayed, displaying in the window the graphic objects of the semantic entities at the (N+1) level that are semantically contained by the semantic entity from the Nth level;

d) displaying a cursor in the display window;

e) receiving a user input to move the cursor toward at least one of the displayed graphic objects for a semantic entity from the (N+1)th level;

f) simulating movement toward a displayed graphic object of a semantic entity from the (N+1)th level by:

(1) increasing the size of the displayed graphic objects of the semantic entities from the (N+1)th level; and

(2) displaying graphic objects of semantic entities at a (N+2)th level contained by the semantic entities from the (N+1)th level;

wherein size of the displayed graphic objects is determined in accordance with the selected information density of the display window.

11. The method of claim 10 wherein the information density is a constant and the size of each graphic object is a function of at least the information density.

* * * * *

US006029195A

United States Patent [19]

Herz

[11] Patent Number: 6,029,195

[45] Date of Patent: Feb. 22, 2000

[54] **SYSTEM FOR CUSTOMIZED ELECTRONIC IDENTIFICATION OF DESIRABLE OBJECTS**

[76] Inventor: **Frederick S. M. Herz**, Box 625 Canaan Valley, Davis, W. Va. 26260

[21] Appl. No.: **08/985,731**

[22] Filed: **Dec. 5, 1997**

Related U.S. Application Data

[63] Continuation-in-part of application No. 08/346,425, Nov. 29, 1994, Pat. No. 5,758,257.

[60] Provisional application No. 60/032,461, Dec. 9, 1996.

[51] **Int. Cl.**[7] **G06F 15/16**; H04H 1/02; H04N 7/14

[52] **U.S. Cl.** **709/219**; 348/1; 455/2; 707/10

[58] **Field of Search** 395/200.47–200.49; 348/1, 2, 6, 7, 8, 10; 455/3.1, 4.1, 4.2, 5.1, 6.1, 6.2; 704/104; 709/217–219, 203; 707/10; H04N 7/10, 7/14, 7/173

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,706,080	11/1987	Sincoskie	340/825.02
5,245,656	9/1993	Loeb et al.	380/23
5,301,109	4/1994	Landauer et al.	364/419.19
5,321,833	6/1994	Chang et al.	395/600
5,331,554	7/1994	Graham	364/419.07
5,331,556	7/1994	Black, Jr. et al.	364/419.08
5,717,923	2/1998	Dedrick	704/104 X
5,724,567	3/1998	Rose et al.	707/10
5,754,939	5/1998	Herz et al.	455/4.2

OTHER PUBLICATIONS

"Scatter/Gather: A Cluster–based Approach to Browsing Large Document Collections" by Cutting et al., 15th Ann Int'l Sigir '92, ACM 318–329.

"Evolving Agents For Personalized Information Filtering", Sheth et al., Proc. 9th IEEE Conference on A1 for Applications.

"A Secure And Privacy–Protecting Protocol For Transmitting Personal Information Between Organizations" Chaum et al.

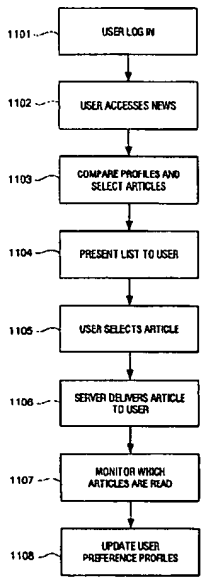
Primary Examiner—John W. Miller

Attorney, Agent, or Firm—Duft, Graziano&Forest,P.C.

[57] **ABSTRACT**

This invention relates to customized electronic identification of desirable objects, such as news articles, in an electronic media environment, and in particular to a system that automatically constructs both a "target profile" for each target object in the electronic media based, for example, on the frequency with which each word appears in an article relative to its overall frequency of use in all articles, as well as a "target profile interest summary" for each user, which target profile interest summary describes the user's interest level in various types of target objects. The system then evaluates the target profiles against the users' target profile interest summaries to generate a user-customized rank ordered listing of target objects most likely to be of interest to each user so that the user can select from among these potentially relevant target objects, which were automatically selected by this system from the plethora of target objects that are profiled on the electronic media. Users' target profile interest summaries can be used to efficiently organize the distribution of information in a large scale system consisting of many users interconnected by means of a communication network. Additionally, a cryptographically-based pseudonym proxy server is provided to ensure the privacy of a user's target profile interest summary, by giving the user control over the ability of third parties to access this summary and to identify or contact the user.

15 Claims, 13 Drawing Sheets



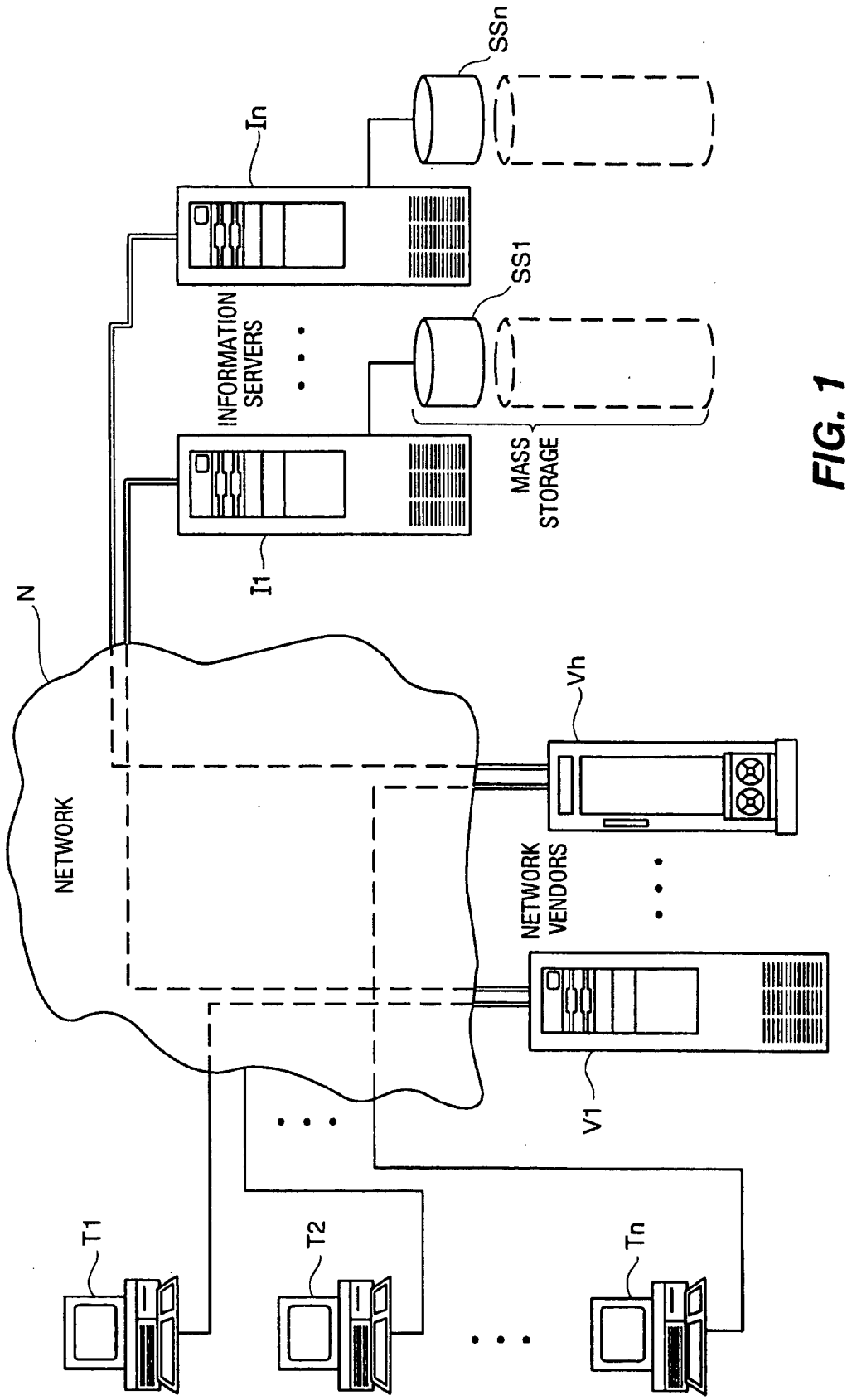


FIG. 1
PRIOR ART

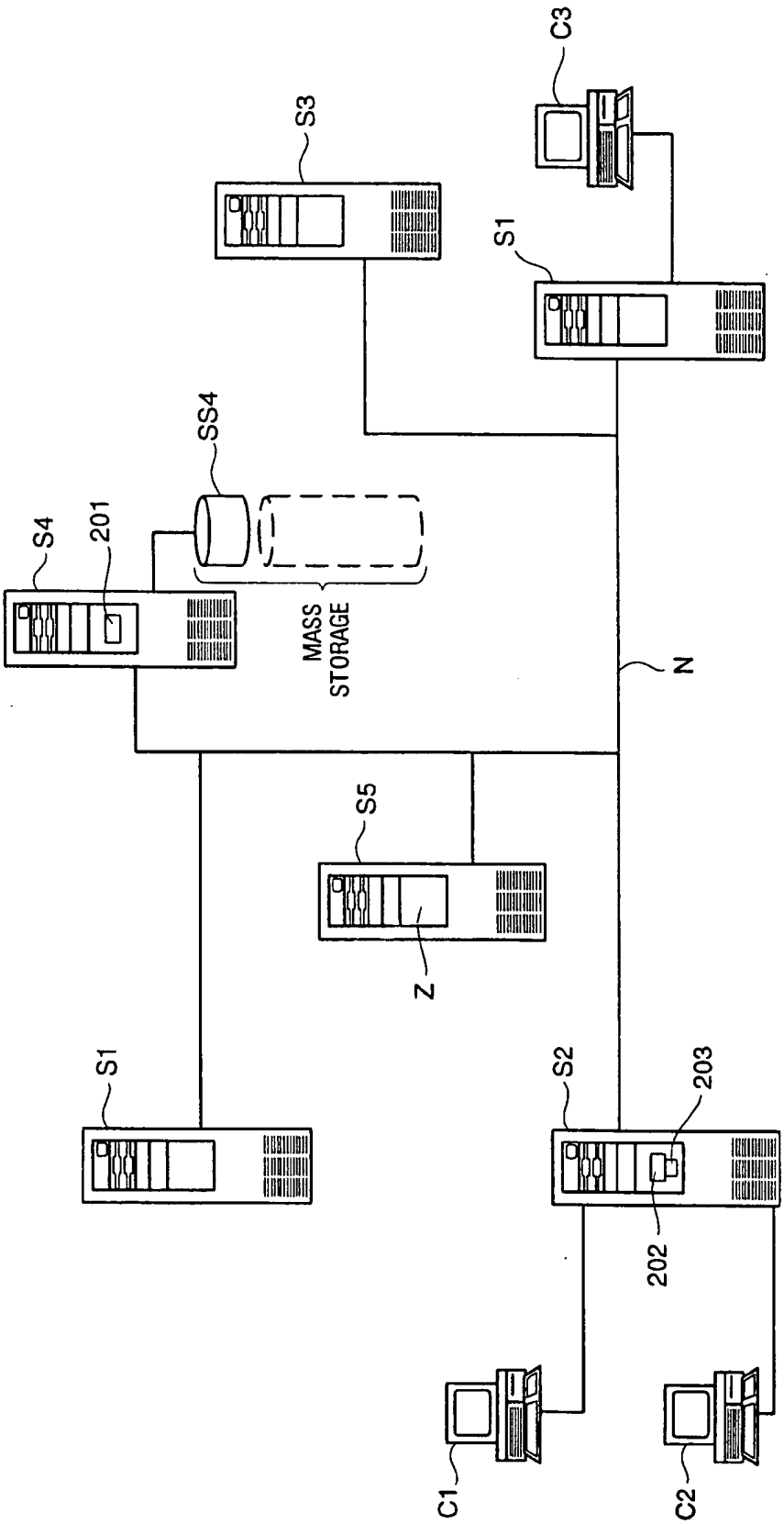


FIG. 2

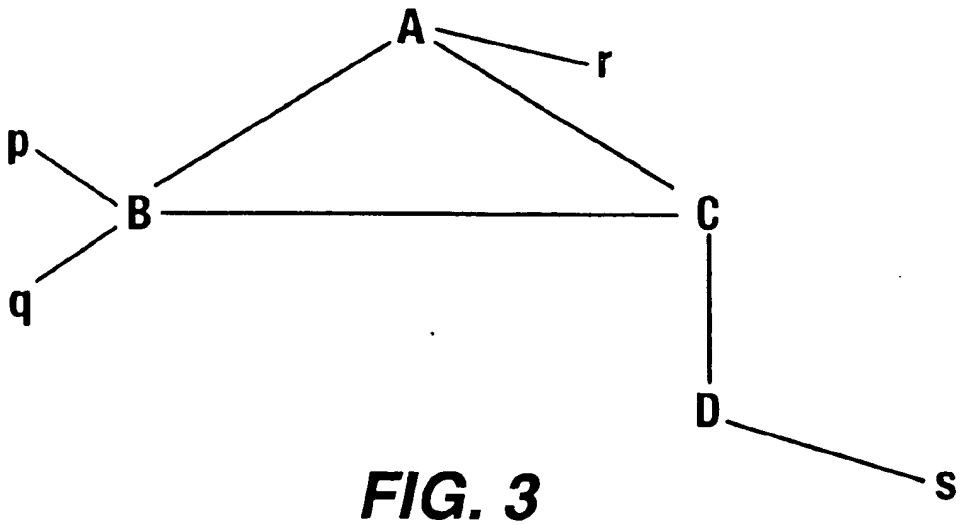


FIG. 3

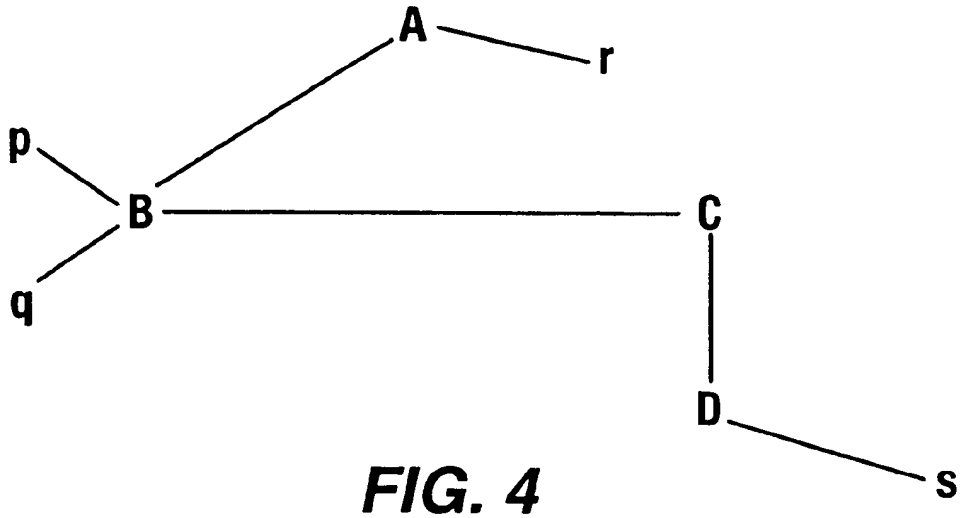


FIG. 4

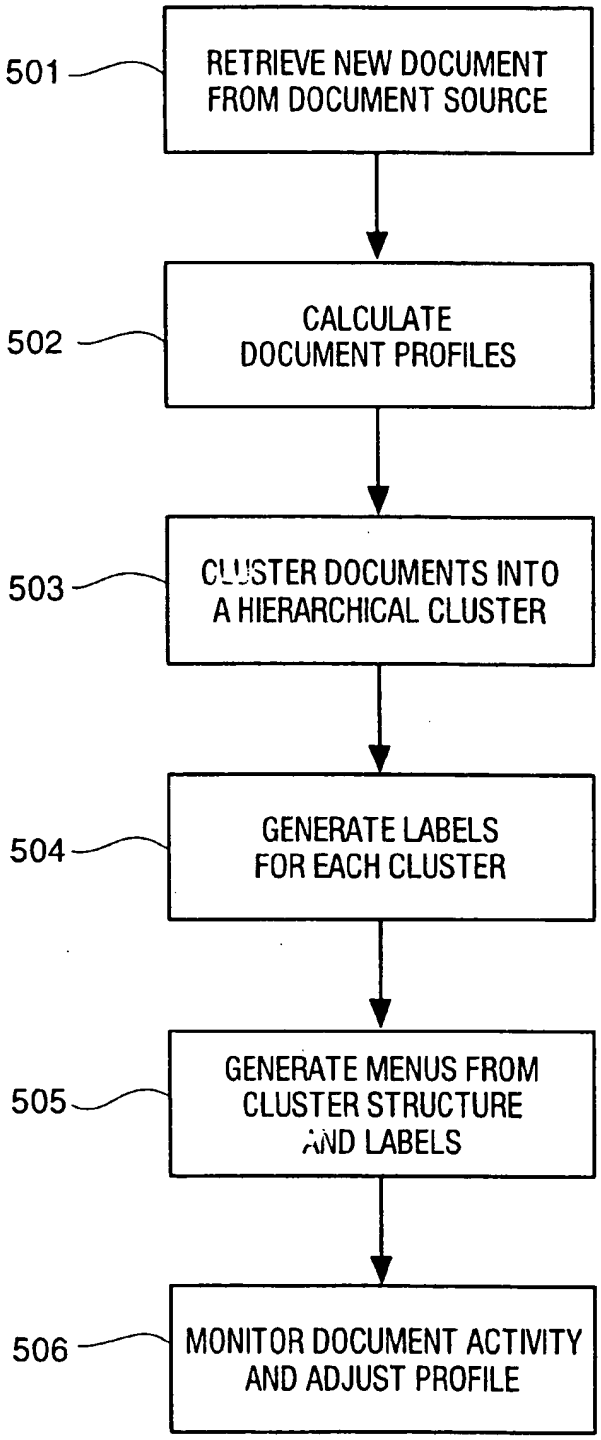
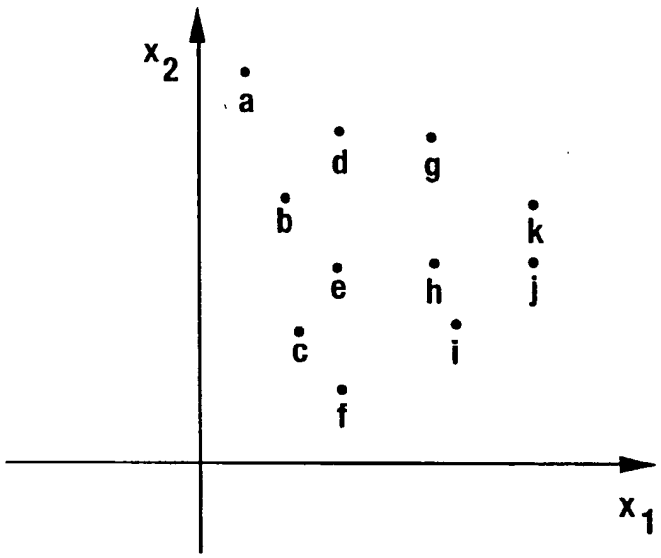

501
RETRIEVE NEW DOCUMENT FROM DOCUMENT SOURCE
502
CALCULATE DOCUMENT PROFILES
503
CLUSTER DOCUMENTS INTO A HIERARCHICAL CLUSTER
504
GENERATE LABELS FOR EACH CLUSTER
505
GENERATE MENUS FROM CLUSTER STRUCTURE AND LABELS
506
MONITOR DOCUMENT ACTIVITY AND ADJUST PROFILE

FIG. 5

FIG. 6


x2
a
d
g
b
k
e
h
j
c
i
f
x1

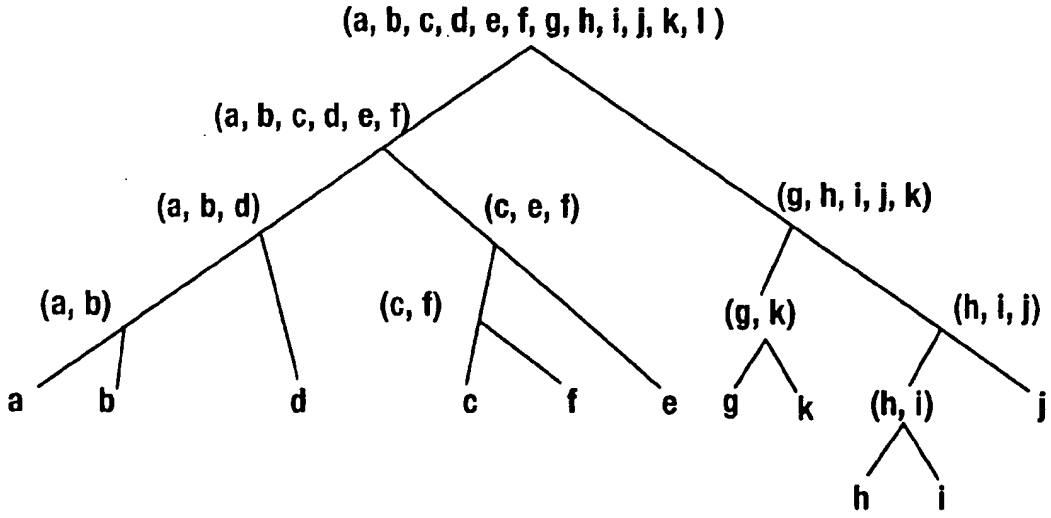

(a, b, c, d, e, f, g, h, i, j, k, l)
(a, b, c, d, e, f)
(a, b, d)
(c, e, f)
(g, h, i, j, k)
(a, b)
(c, f)
(g, k)
(h, i, j)
a
b
d
c
f
e
g
k
(h, i)
j
h
i

FIG. 7

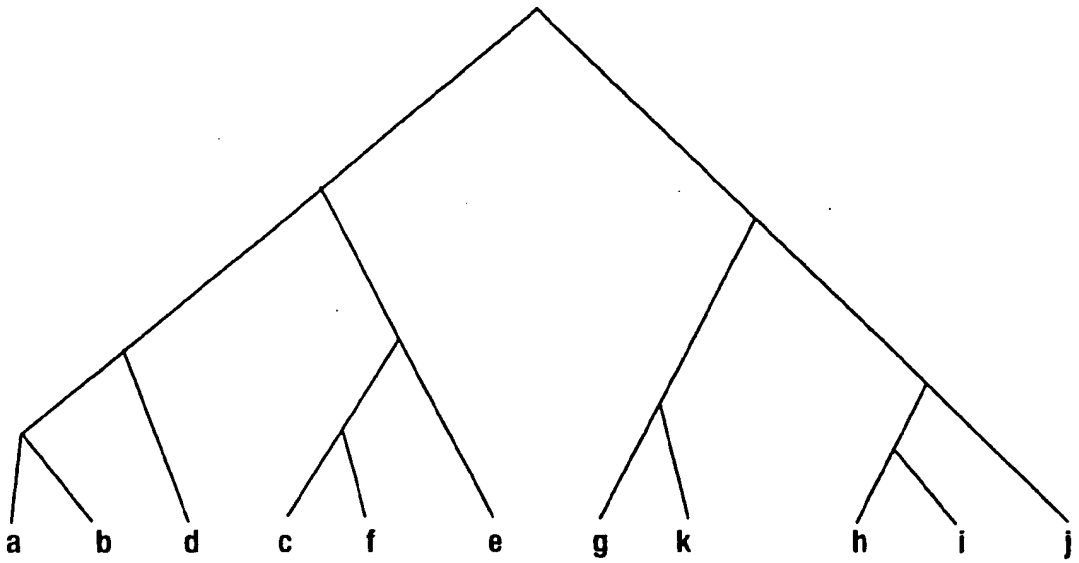


FIG. 8

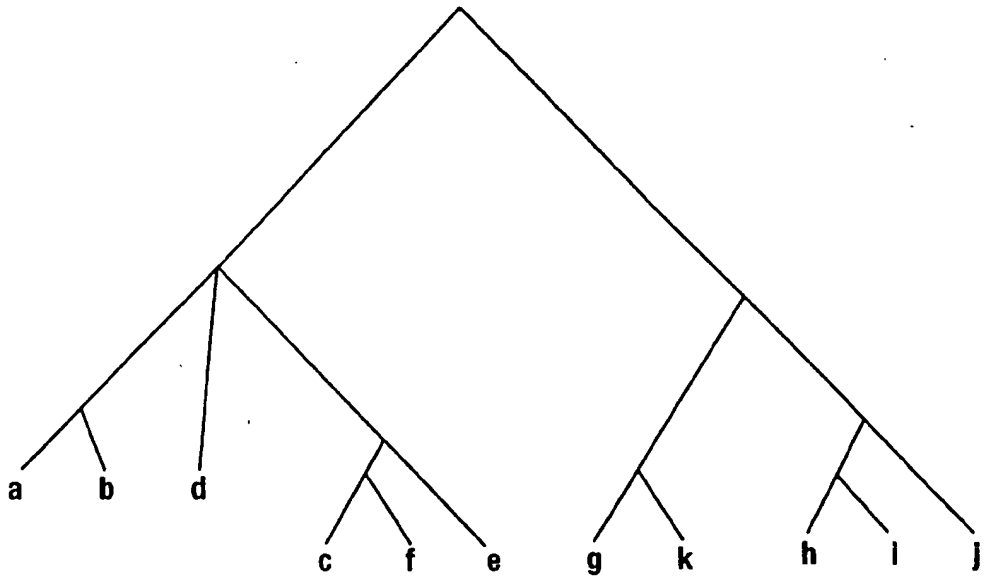


FIG. 9

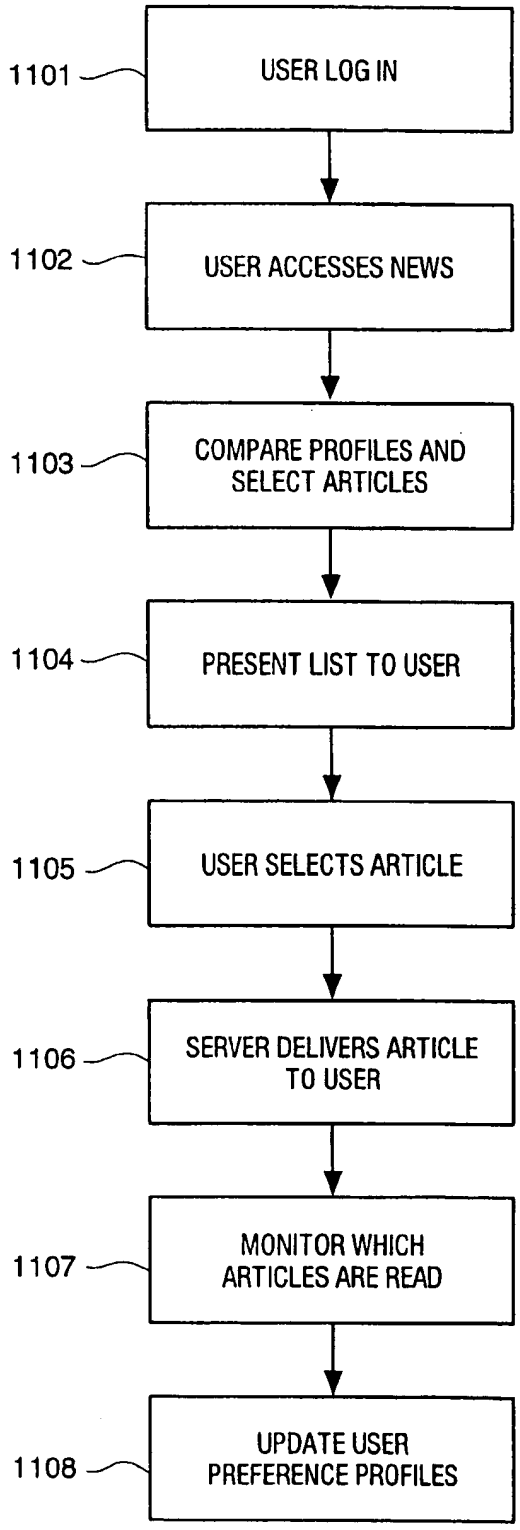

1101
USER LOG IN
1102
USER ACCESSES NEWS
1103
COMPARE PROFILES AND SELECT ARTICLES
1104
PRESENT LIST TO USER
1105
USER SELECTS ARTICLE
1106
SERVER DELIVERS ARTICLE TO USER
1107
MONITOR WHICH ARTICLES ARE READ
1108
UPDATE USER PREFERENCE PROFILES

FIG. 10

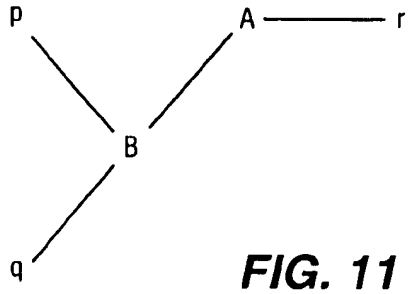

p
A
r
B
q

FIG. 11

FIG. 12

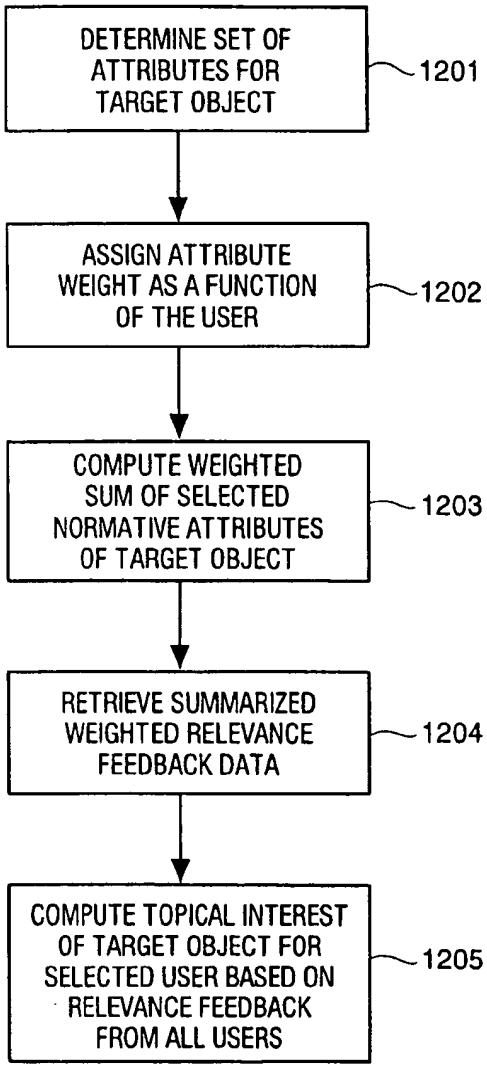

DETERMINE SET OF ATTRIBUTES FOR TARGET OBJECT
1201
ASSIGN ATTRIBUTE WEIGHT AS A FUNCTION OF THE USER
1202
COMPUTE WEIGHTED SUM OF SELECTED NORMATIVE ATTRIBUTES OF TARGET OBJECT
1203
RETRIEVE SUMMARIZED WEIGHTED RELEVANCE FEEDBACK DATA
1204
COMPUTE TOPICAL INTEREST OF TARGET OBJECT FOR SELECTED USER BASED ON RELEVANCE FEEDBACK FROM ALL USERS
1205

FIG. 13A

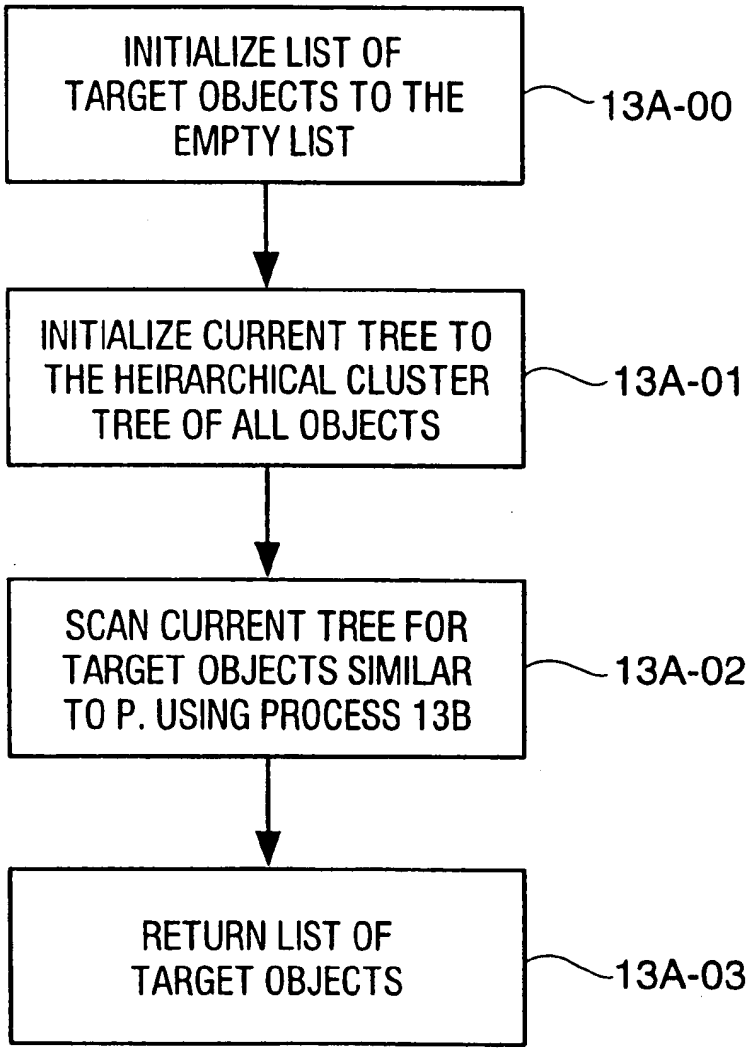

INITIALIZE LIST OF TARGET OBJECTS TO THE EMPTY LIST
13A-00
INITIALIZE CURRENT TREE TO THE HEIRARCHICAL CLUSTER TREE OF ALL OBJECTS
13A-01
SCAN CURRENT TREE FOR TARGET OBJECTS SIMILAR TO P. USING PROCESS 13B
13A-02
RETURN LIST OF TARGET OBJECTS
13A-03

FIG. 13B

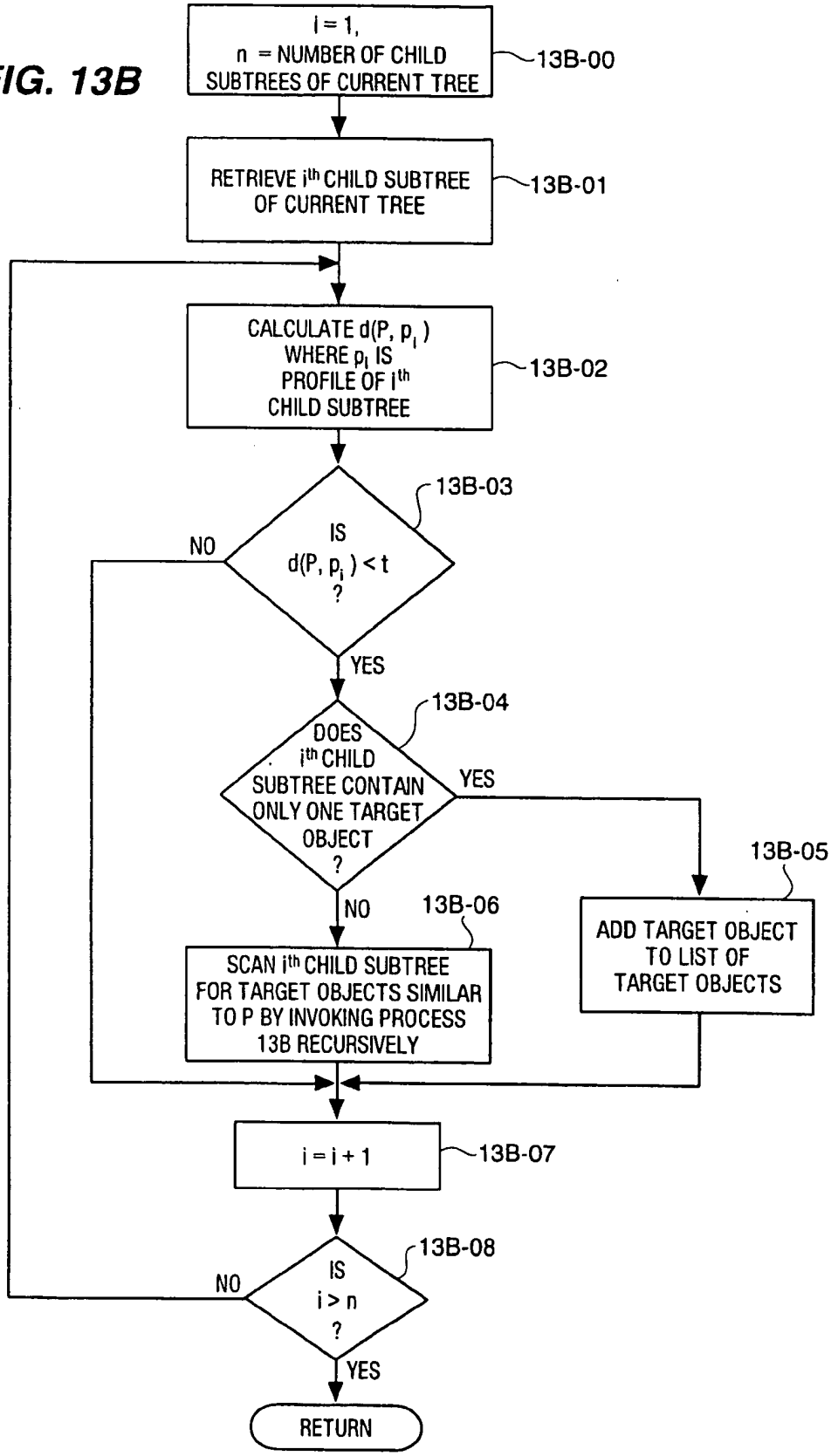

i = 1,
n = NUMBER OF CHILD SUBTREES OF CURRENT TREE
13B-00
RETRIEVE ith CHILD SUBTREE OF CURRENT TREE
13B-01
CALCULATE d(P, p_i) WHERE p_i IS PROFILE OF ith CHILD SUBTREE
13B-02
IS d(P, p_i) < t ?
13B-03
NO
YES
DOES ith CHILD SUBTREE CONTAIN ONLY ONE TARGET OBJECT ?
13B-04
YES
NO
13B-05
ADD TARGET OBJECT TO LIST OF TARGET OBJECTS
13B-06
SCAN ith CHILD SUBTREE FOR TARGET OBJECTS SIMILAR TO P BY INVOKING PROCESS 13B RECURSIVELY
i = i + 1
13B-07
IS i > n ?
13B-08
NO
YES
RETURN

FIG. 14

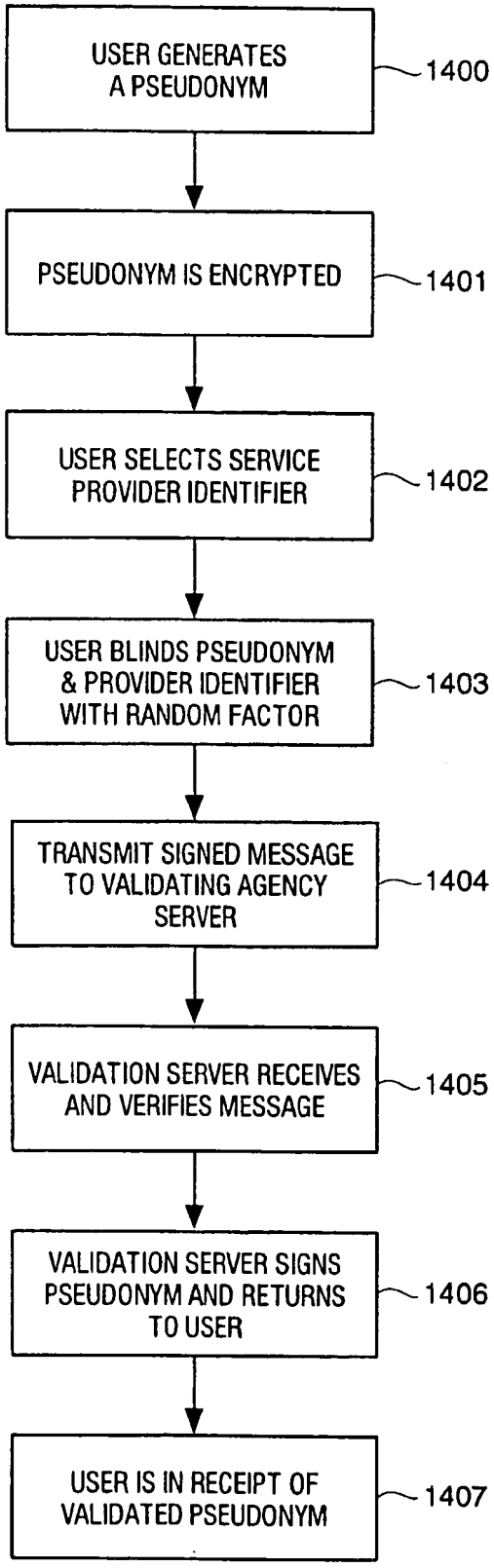

USER GENERATES A PSEUDONYM
1400
PSEUDONYM IS ENCRYPTED
1401
USER SELECTS SERVICE PROVIDER IDENTIFIER
1402
USER BLINDS PSEUDONYM & PROVIDER IDENTIFIER WITH RANDOM FACTOR
1403
TRANSMIT SIGNED MESSAGE TO VALIDATING AGENCY SERVER
1404
VALIDATION SERVER RECEIVES AND VERIFIES MESSAGE
1405
VALIDATION SERVER SIGNS PSEUDONYM AND RETURNS TO USER
1406
USER IS IN RECEIPT OF VALIDATED PSEUDONYM
1407

FIG. 15

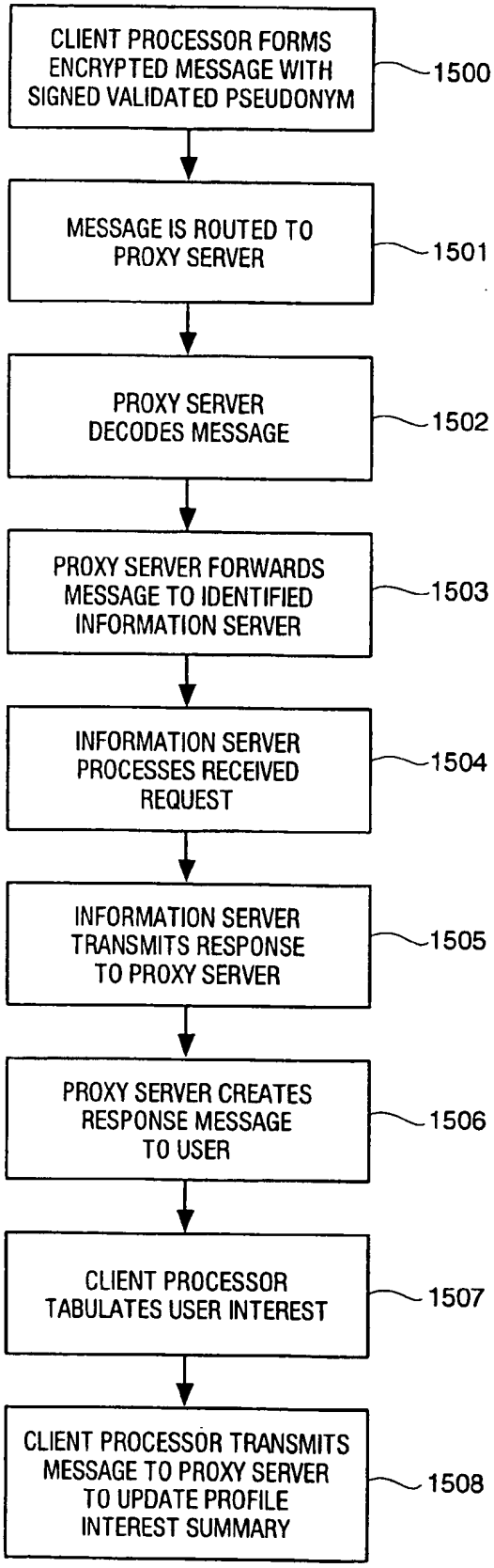
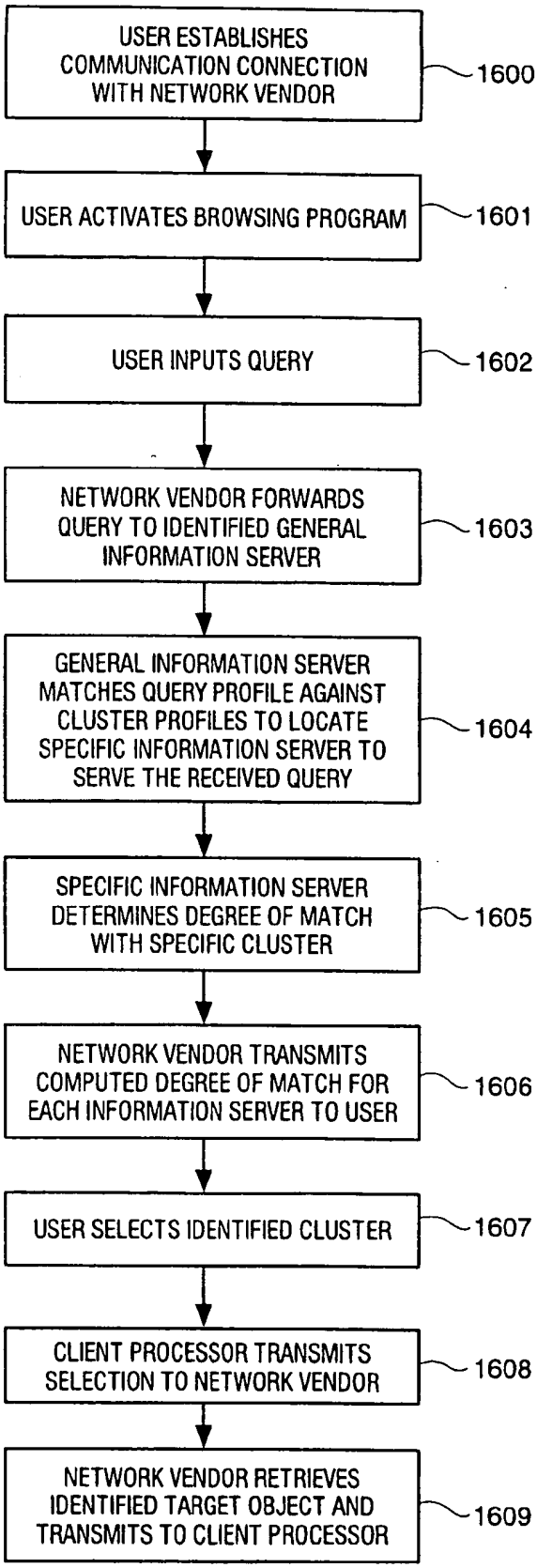

CLIENT PROCESSOR FORMS ENCRYPTED MESSAGE WITH SIGNED VALIDATED PSEUDONYM
1500
MESSAGE IS ROUTED TO PROXY SERVER
1501
PROXY SERVER DECODES MESSAGE
1502
PROXY SERVER FORWARDS MESSAGE TO IDENTIFIED INFORMATION SERVER
1503
INFORMATION SERVER PROCESSES RECEIVED REQUEST
1504
INFORMATION SERVER TRANSMITS RESPONSE TO PROXY SERVER
1505
PROXY SERVER CREATES RESPONSE MESSAGE TO USER
1506
CLIENT PROCESSOR TABULATES USER INTEREST
1507
CLIENT PROCESSOR TRANSMITS MESSAGE TO PROXY SERVER TO UPDATE PROFILE INTEREST SUMMARY
1508

FIG. 16


USER ESTABLISHES COMMUNICATION CONNECTION WITH NETWORK VENDOR
1600
USER ACTIVATES BROWSING PROGRAM
1601
USER INPUTS QUERY
1602
NETWORK VENDOR FORWARDS QUERY TO IDENTIFIED GENERAL INFORMATION SERVER
1603
GENERAL INFORMATION SERVER MATCHES QUERY PROFILE AGAINST CLUSTER PROFILES TO LOCATE SPECIFIC INFORMATION SERVER TO SERVE THE RECEIVED QUERY
1604
SPECIFIC INFORMATION SERVER DETERMINES DEGREE OF MATCH WITH SPECIFIC CLUSTER
1605
NETWORK VENDOR TRANSMITS COMPUTED DEGREE OF MATCH FOR EACH INFORMATION SERVER TO USER
1606
USER SELECTS IDENTIFIED CLUSTER
1607
CLIENT PROCESSOR TRANSMITS SELECTION TO NETWORK VENDOR
1608
NETWORK VENDOR RETRIEVES IDENTIFIED TARGET OBJECT AND TRANSMITS TO CLIENT PROCESSOR
1609

SYSTEM FOR CUSTOMIZED ELECTRONIC IDENTIFICATION OF DESIRABLE OBJECTS

CROSS-REFERENCE TO RELATED APPLICATIONS

This patent application was originally filed as Provisional Patent Application Ser. No. 60/032,461 on Dec. 9, 1996 and is a continuation-in-part of U.S. patent application Ser. No. 08/346,425, filed Nov. 29, 1994, now U.S. Pat. No. 5,758,257 and titled "SYSTEM AND METHOD FOR SCHEDULING BROADCAST OF AND ACCESS TO VIDEO PROGRAMS AND OTHER DATA USING CUSTOMER PROFILES", which application is assigned to the same assignee as the present application.

FIELD OF INVENTION

This invention relates to customized electronic identification of desirable objects, such as news articles, in an electronic media environment, and in particular to a system that automatically constructs both a "target profile" for each target object in the electronic media based, for example, on the frequency with which each word appears in an article relative to its overall frequency of use in all articles, as well as a "target profile interest summary" for each user, which target profile interest summary describes the user's interest level in various types of target objects. The system then evaluates the target profiles against the users' target profile interest summaries to generate a user-customized rank ordered listing of target objects most likely to be of interest to each user so that the user can select from among these potentially relevant target objects, which were automatically selected by this system from the plethora of target objects that are profiled on the electronic media. Users' target profile interest summaries can be used to efficiently organize the distribution of information in a large scale system consisting of many-users interconnected by means of a communication network. Additionally, a cryptographically based proxy server is provided to ensure the privacy of a user's target profile interest summary, by giving the user control over the ability of third parties to access this summary and to identify or contact the user.

PROBLEM

It is a problem in the field of electronic media to enable a user to access information of relevance and interest to the user without requiring the user to expend an excessive amount of time and energy searching for the information. Electronic media, such as on-line information sources, provide a vast amount of information to users, typically in the form of "articles," each of which comprises a publication item or document that relates to a specific topic. The difficulty with electronic media is that the amount of information available to the user is overwhelming and the article repository systems that are connected on-line are not organized in a manner that sufficiently simplifies access to only the articles of interest to the user. Presently, a user either fails to access relevant articles because they are not easily identified or expends a significant amount of time and energy to conduct an exhaustive search of all articles to identify those most likely to be of interest to the user. Furthermore, even if the user conducts an exhaustive search, present information searching techniques do not necessarily accurately extract only the most relevant articles, but also present articles of marginal relevance due to the functional limitations of the information searching techniques. There is also no existing system which automatically estimates the inherent quality of an article or other target object to distinguish among a number of articles or target objects identified as of possible interest to a user.

Therefore, in the field of information retrieval, there is a long-standing need for a system which enables users to navigate through the plethora of information. With commercialization of communication networks, such as the Internet, the growth of available information has increased. Customization of the information delivery process to the user's unique tastes and interests is the ultimate solution to this problem. However, the techniques which have been proposed to date either only address the user's interests on a superficial level or provide greater depth and intelligence at the cost of unwanted demands on the user's time and energy. While many researchers have agreed that traditional methods have been lacking in this regard, no one to date has successfully addressed these problems in a holistic manner and provided a system that can fully learn and reflect the user's tastes and interests. This is particularly true in a practical commercial context, such as on-line services available on the Internet. There is a need for an information retrieval system that is largely or entirely passive, unobtrusive, undemanding of the user, and yet both precise and comprehensive in its ability to learn and truly represent the user's tastes and interests. Present information retrieval systems require the user to specify the desired information retrieval behavior through cumbersome interfaces.

Users may receive information on a computer network either by actively retrieving the information or by passively receiving information that is sent to them. Just as users of information retrieval systems face the problem of too much information, so do users who are targeted with electronic junk mail by individuals and organizations. An ideal system would protect the user from unsolicited advertising, both by automatically extracting only the most relevant messages received by electronic mail, and by preserving the confidentiality of the user's preferences, which should not be freely available to others on the network.

Researchers in the field of published article information retrieval have devoted considerable effort to finding efficient and accurate methods of allowing users to select articles of interest from a large set of articles. The most widely used methods of information retrieval are based on keyword matching: the user specifies a set of keywords which the user thinks are exclusively found in the desired articles and the information retrieval computer retrieves all articles which contain those keywords. Such methods are fast, but are notoriously unreliable, as users may not think of the right keywords, or the keywords may be used in unwanted articles in an irrelevant or unexpected context. As a result, the information retrieval computers retrieve many articles which are unwanted by the user. The logical combination of keywords and the use of wild-card search parameters help improve the accuracy of keyword searching but do not completely solve the problem of inaccurate search results. Starting in the 1960's, an alternate approach to information retrieval was developed: users were presented with an article and asked if it contained the information they wanted, or to quantify how close the information contained in the article was to what they wanted. Each article was described by a profile which comprised either a list of the words in the article or, in more advanced systems, a table of word frequencies in the article. Since a measure of similarity between articles is the distance between their profiles, the measured similarity of article profiles can be used in article retrieval. For example, a user searching for information on a subject can write a short description of the desired infor-

mation. The information retrieval computer generates an article profile for the request and then retrieves articles with profiles similar to the profile generated for the request. These requests can then be refined using "relevance feedback", where the user actively or passively rates the articles retrieved as to how close the information contained therein is to what is desired. The information retrieval computer then uses this relevance feedback information to refine the request profile and the process is repeated until the user either finds enough articles or tires of the search.

A number of researchers have looked at methods for selecting articles of most interest to users. An article titled "Social Information filtering: algorithms for automating 'word of mouth'" was published at the CHi-95 Proceedings by Patti Maes et al and describes the Ringo information retrieval system which recommends musical selections. The Ringo system requires active feedback from the users—users must manually specify how much they like or dislike each musical selection. The Ringo system maintains a complete list of users ratings of music selections and makes recommendations by finding which selections were liked by multiple people. However, the Ringo system does not take advantage of any available descriptions of the music, such as structured descriptions in a data base, or free text, such as that contained in music reviews. An article titled "Evolving agents for personalized information filtering", published at the Proc. 9th IEEE Conf. on AI for Applications by Sheth and Maes, described the use of agents for information filtering which use genetic algorithms to learn to categorize Usenet news articles. In this system, users must define news categories and the users actively indicate their opinion of the selected articles. Their system uses a list of keywords to represent sets of articles and the records of users' interests are updated using genetic algorithms.

A number of other research groups have looked at the automatic generation and labeling of clusters of articles for the purpose of browsing through the articles. A group at Xerox Parc published a paper titled "Scatter/gather: a cluster-based approach to browsing large article collections" at the 15 Ann. Int'l SIGIR '92, ACM 318–329 (Cutting et al. 1992). This group developed a method they call "scatter/gather" for performing information retrieval searches. In this method, a collection of articles is "scattered" into a small number of clusters, the user then chooses one or more of these clusters based on short summaries of the cluster. The selected clusters are then "gathered" into a subcollection, and then the process is repeated. Each iteration of this process is expected to produce a small, more focused collection. The cluster "summaries" are generated by picking those words which appear most frequently in the cluster and the titles of those articles closest to the center of the cluster. However, no feedback from users is collected or stored, so no performance improvement occurs over time.

Apple's Advanced Technology Group has developed an interface based on the concept of a "pile of articles". This interface is described in an article titled "A 'pile' metaphor for supporting casual organization of information in Human factors in computer systems" published in CHI '92 Conf. Proc. 627–634 by Mander, R. G. Salomon and Y. Wong. 1992. Another article titled "Content awareness in a file system interface: implementing the 'pile' metaphor for organizing information" was published in 16 Ann. Int'l SIGIR '93, ACM 260–269 by Rose E. D. et al. The Apple interface uses word frequencies to automatically file articles by picking the pile most similar to the article being filed. This system functions to cluster articles into subpiles, determine key words for indexing by picking the words with the largest TF/IDF (where TF is term (word) frequency and IDF is the inverse document frequency) and label piles by using the determined key words.

Numerous patents address information retrieval methods, but none develop records of a user's interest based on passive monitoring of which articles the user accesses. None of the systems described in these patents pre sent computer architectures to allow fast retrieval of articles distributed across many computers. None of the systems described in these patents address issues of using such article retrieval and matching methods for purposes of commerce or of matching users with common interests or developing records of users' interests. U.S. Pat. No. 5,321,833 issued to Chang et al. teaches a method in which users choose terms to use in an information retrieval query, and specify the relative weightings of the different terms. The Chang system then calculates multiple levels of weighting criteria. U.S. Pat. No. 5,301,109 issued to Landauer et al. teaches a method for retrieving articles in a multiplicity of languages by constructing "latent vectors" (SVD or PCA vectors) which represent correlations between the different words. U.S. Pat. No. 5,331,554 issued to Graham et al. discloses a method for retrieving segments of a manual by comparing a query with nodes in a decision tree. U.S. Pat. No. 5,331,556 addresses techniques for deriving morphological part-of-speech information and thus to make use of the similarities of different forms of the same word (e.g. "article" and "articles").

Therefore, there presently is no information retrieval and delivery system operable in an electronic media environment that enables a user to access information of relevance and interest to the user without requiring the user to expend an excessive amount of time and energy.

SOLUTION

The above-described problems are solved and a technical advance achieved in the field by the system for customized electronic identification of desirable objects in an electronic media environment, which system enables a user to access target objects of relevance and interest to the user without requiring the user to expend an excessive amount of time and energy. Profiles of the target objects are stored on electronic media and are accessible via a data communication network. In many applications, the target objects are informational in nature, and so may themselves be stored on electronic media and be accessible via a data communication network.

Relevant definitions of terms for the purpose of this description include: (a.) an object available for access by the user, which may be either physical or electronic in nature, is termed a "target object", (b.) a digitally represented profile indicating that target object's attributes is termed a "target profile", (c.) the user looking for the target object is termed a "user", (d.) a profile holding that user's attributes, including age/zip code/etc. is termed a "user profile", (e.) a summary of digital profiles of target objects that a user likes and/or dislikes, is termed the "target profile interest summary" of that user, (f.) a profile consisting of a collection of attributes, such that a user likes target objects whose profiles are similar to this collection of attributes, is termed a "search profile" or in some contexts a "query" or "query profile," (g.) a specific embodiment of the target profile interest summary which comprises a set of search profiles is termed the "search profile set" of a user, (h.) a collection of target objects with similar profiles, is termed a "cluster," (i.) an aggregate profile formed by averaging the attributes of all tar get objects in a cluster, termed a "cluster profile," (j.) a real

number determined by calculating the statistical variance of the profiles of all target objects in a cluster, is termed a "cluster variance," (k.) a real number determined by calculating the maximum distance between the profiles of any two target objects in a cluster, is termed a "cluster diameter."

The system for electronic identification of desirable objects of the present invention automatically constructs both a target profile for each target object in the electronic media based, for example, on the frequency with which each word appears in an article relative to its overall frequency of use in all articles, as well as a "target profile interest summary" for each user, which target profile interest summary describes the user's interest level in various types of target objects. The system then evaluates the target profiles against the users' target profile interest summaries to generate a user-customized rank ordered listing of target objects most likely to be of interest to each user so that the user can select from among these potentially relevant target objects, which were automatically selected by this system from the plethora of target objects available on the electronic media.

Because people have multiple interests, a target profile interest summary for a single user must represent multiple areas of interest, for example, by consisting of a set of individual search profiles, each of which identifies one of the user's areas of interest. Each user is presented with those target objects whose profiles most closely match the user's interests as described by the user's target profile interest summary. Users' target profile interest summaries are automatically updated on a continuing basis to reflect each user's changing interests. In addition, target objects can be grouped into clusters based on their similarity to each other, for example, based on similarity of their topics in the case where the target objects are published articles, and menus automatically generated for each cluster of target objects to allow users to navigate throughout the clusters and manually locate target objects of interest. For reasons of confidentiality and privacy, a particular user may not wish to make public all of the interests recorded in the user's target profile interest summary, particularly when these interests are determined by the user's purchasing patterns. The user may desire that all or part of the target profile interest summary be kept confidential, such as information relating to the user's political, religious, financial or purchasing behavior; indeed, confidentiality with respect to purchasing behavior is the user's legal right in many states. It is therefore necessary that data in a user's target profile interest summary be protected from unwanted disclosure except with the user's agreement. At the same time, the user's target profile interest summaries must be accessible to the relevant servers that perform the matching of target objects to the users, if the benefit of this matching is desired by both providers and consumers of the target objects. The disclosed system provides a solution to the privacy problem by using a proxy server which acts as an intermediary between the information provider and the user. The proxy server dissociates the user's true identity from the pseudonym by the use of cryptographic techniques. The proxy server also permits users to control access to their target profile interest summaries and/or user profiles, including provision of this information to marketers and advertisers if they so desire, possibly in exchange for cash or other considerations. Marketers may purchase these profiles in order to target advertisements to particular users, or they may purchase partial user profiles, which do not include enough information to identify the individual users in question, in order to carry out standard kinds of demographic analysis and market research on the resulting database of partial user profiles. Pseudonymous control of an information server suggests how a special discount can be issued to a user's pseudonym and that such a digital credential is provided to the user as a result of his/her user profile making him/her eligible. The user may thus present this type of credential to the appropriate vendor to take advantage of the discount. This technique can be extended also to smart cards wherein the digital credential providing the discount is downloaded from the client to the smart card and upon presentation, the vendor may if desired, delete the credential upon redemption by the user. These discount credentials may similarly include any of the discount types (customized promotions) herein disclosed wherein each purchase may identified (characterized) and credentialized by the vendor onto the user's smart card and/or the vendor's system.

In the preferred embodiment of the invention, the system for customized electronic identification of desirable objects uses a fundamental methodology for accurately and efficiently matching users and target objects by automatically calculating, using and updating profile information that describes both the users' interests and the target objects' characteristics. The target objects may be published articles, purchasable items, or even other people, and their properties are stored, and/or represented and/or denoted on the electronic media as (digital) data. Examples of target objects can include, but are not limited to: a newspaper story of potential interest, a movie to watch, an item to buy, e-mail to receive, or another person to correspond with. In one suggested application, the user is a sender of email (which may have originated from the user for or from another external source such as from outside of a large organization) and the target objects are users who might be considered most appropriate based upon previous messages which they have received, read and responded to. Accordingly, like other target objects, users (or user pseudonyms) in accordance with their user profiles (or portions of which they have disclosed) may be organized and browsed within an automatically generated menu tree, which is below described in detail. In all these cases, the information delivery process in the preferred embodiment is based on determining the similarity between a profile for the target object and the profiles of target objects for which the user (or a similar user) has provided positive feedback in the past. The individual data that describe a target object and constitute the target object's profile are herein termed "attributes" of the target object. Attributes may include, but are not limited to, the following: (1) long pieces of text (a newspaper story, a movie review, a product description or an advertisement), (2) short pieces of text (name of a movie's director, name of town from which an advertisement was placed, name of the language in which an article was written), (3) numeric measurements (price of a product, rating given to a movie, reading level of a book), (4) associations with other types of objects (list of actors in a movie, list of persons who have read a document). Any of these attributes, but especially the numeric ones, may correlate with the quality of the target object, such as measures of its popularity (how often it is accessed) or of user satisfaction (number of complaints received).

The preferred embodiment of the system for customized electronic identification of desirable objects operates in an electronic media environment for accessing these target objects, which may be news, electronic mail, other published documents, or product descriptions. The system in its broadest construction comprises three conceptual modules, which may be separate entities distributed across many implementing systems, or combined into a lesser subset of physical entities. The specific embodiment of this system

disclosed herein illustrates the use of a first module which automatically constructs a "target profile" for each target object in the electronic media based on various descriptive attributes of the target object. A second module uses interest feedback from users to construct a "target profile interest summary" for each user, for example in the form of a "search profile set" consisting of a plurality of search profiles, each of which corresponds to a single topic of high interest for the user. The system further includes a profile processing module which estimates each user's interest in various target objects by reference to the users' target profile interest summaries, for example by comparing the target profiles of these target objects against the search profiles in users' search profile sets, and generates for each user a customized rank-ordered listing of target objects most likely to be of interest to that user. Each user's target profile interest summary is automatically updated on a continuing basis to reflect the user's changing interests.

Target objects may be of various sorts, and it is sometimes advantageous to use a single system that delivers and/or clusters target objects of several distinct sorts at once, in a unified framework. For example, users who exhibit a strong interest in certain novels may also show an interest in certain movies, presumably of a similar nature. A system in which some target objects are novels and other target objects are movies can discover such a correlation and exploit it in order to group particular novels with particular movies, e.g., for clustering purposes, or to recommend the movies to a user who has demonstrated interest in the novels. Similarly, if users who exhibit an interest in certain World Wide Web sites also exhibit an interest in certain products, the system can match the products with the sites and thereby recommend to the marketers of those products that they place advertisements at those sites, e.g., in the form of hypertext links to their own sites. The presently described system explains the techniques for target advertising (on a user by user basis) through both links from advertisements on a web page which tends to be visited by the most likely buyers of that particular product or service, and routing advertisements to such users via email. (This assumes that be cause user visitorship is measured at the level of the web page, certain pages within the web site may be more appropriate for certain advertisements due to the slight differences in its visitorship. Text chat(or acoustic voice chat) using a text to speech conversion module may be used in conjunction with real time profiling of the real time user dialogues occurring within that chat session. Advertisements which are relevant nature of the content being discussed at present may provide temporary links to the appropriate product such that when the nature of the content changes the advertisements changes (may disappear) accordingly.

The ability to measure the similarity of profiles describing target objects and a user's interests can be applied in two basic ways: filtering and browsing. Filtering is useful when large numbers of target objects are described in the electronic medias pace. These target objects can for example be articles that are received or potentially received by a user, who only has time to read a small fraction of them. For example, one might potentially receive all items on the AP news wire service, all items posted to a number of news groups, all advertisements in a set of newspapers, or all unsolicited electronic mail, but few people have the time or inclination to read so many articles. A filtering system in the system for customized electronic identification of desirable objects automatically selects a set of articles that the user is likely to wish to read. The accuracy of this filtering system improves over time by noting which articles the user reads and by generating a measurement of the depth to which the user reads each article. This information is then used to update the user's target profile interest summary. Browsing provides an alternate method of selecting a small subset of a large number of target objects, such as articles. Articles are organized so that users can actively navigate among groups of articles by moving from one group to a larger, more general group, to a smaller, more specific group, or to a closely related group. Each individual article forms a one-member group of its own, so that the user can navigate to and from individual article s as well as larger groups. The methods used by the system for customized electronic identification of desirable objects allow articles to be grouped into clusters and the clusters to be grouped and merged into larger and larger clusters. These hierarchies of clusters then form the basis for menuing and navigational systems to allow the rapid searching of large numbers of articles. This same clustering technique is applicable to any type of target objects that can be profiled on the electronic media such as product selections within a menu or throughout the World Wide Web.

There are a number of variations on the theme of developing and using profiles for article retrieval. Variations of this basic system are disclosed and comprise a system to filter electronic mail, an extension for retrieval of target objects such as purchasable items which may have more complex descriptions, a system to automatically build and alter menuing systems for browsing and searching through large numbers of target objects, and a system to construct virtual communities of people with common interests. These intelligent filters and browsers are necessary to provide a truly passive, intelligent system interface. A user interface that permits intuitive browsing and filtering represents for the first time an intelligent system for determining the affinities between users and target objects. The detailed, comprehensive target profiles and user-specific target profile interest summaries enable the system to provide responsive routing of specific queries for user information access. The information maps so produced and the application of users' target profile interest summaries to predict the information consumption patterns of a user allows for pre-caching of data at locations on the data communication network and at times that minimize the traffic flow in the communication network to thereby efficiently provide the desired information to the user and/or conserve valuable storage space by only storing those target objects (or segments thereof) which are relevant to the user's interests.

BRIEF DESCRIPTION OF THE DRAWING

FIG. **1** illustrates in block diagram form a typical architecture of an electronic media system in which the system for customized electronic identification of desirable objects of the present invention can be implemented as part of a user server system;

FIG. **2** illustrates in block diagram form one embodiment of the system for customized electronic identification of desirable objects;

FIGS. **3** and **4** illustrate typical network trees;

FIG. **5** illustrates in flow diagram form a method for automatically generating article profiles and an associated hierarchical menu system;

FIGS. **6–9** illustrate examples of menu generating process;

FIG. **10** illustrates in flow diagram form the operational steps taken by the system for customized electronic identification of desirable objects to screen articles for a user;

FIG. 11 illustrates a hierarchical cluster tree example;

FIG. 12 illustrates in flow diagram form the process for determination of likelihood of interest by a specific user in a selected target object;

FIGS. 13A–B illustrate in flow diagram form the automatic clustering process;

FIG. 14 illustrates in flow diagram form the use of the pseudonymous server;

FIG. 15 illustrates in flow diagram form the use of the system for accessing information in response to a user query; and

FIG. 16 illustrates in flow diagram form the use of the system for accessing information in response to a user query when the system is a distributed network implementation.

DETAILED DESCRIPTION

MEASURING SIMILARITY

This section describes a general procedure for automatically measuring the similarity between two target objects, or, more precisely, between target profiles that are automatically generated for each of the two target objects. This similarity determination process is applicable to target objects in a wide variety of contexts. Target objects being compared can be, as an example but not limited to: textual documents, human beings, movies, or mutual funds. It is assumed that the target profiles which describe the target objects are stored at one or more locations in a data communication network on data storage media associated with a computer system.

The computed similarity measurements serve as input to additional processes, which function to enable human users to locate desired target objects using a large computer system. These additional processes estimate a human user's interest in various target objects, or else cluster a plurality of target objects in to logically coherent groups. The methods used by these additional processes might in principle be implemented on either a single computer or on a computer network. Jointly or separately, they form the underpinning for various sorts of database systems and information retrieval systems.

Target Objects and Attributes

In classical Information Retrieval (IR) technology, the user is a literate human and the target objects in question are textual documents stored on data storage devices interconnected to the user via a computer network. That is, the target objects consist entirely of text, and so are digitally stored on the data storage devices within the computer network. However, there are other target object domains that present related retrieval problems that are not capable of being solved by present information retrieval technology which are applicable to targeting of articles and advertisements to readers of an on-line newspaper:

(a.) the user is a film buff and the target objects are movies available on videotape.

(b.) the user is a consumer and the target objects are used cars being sold.

(c.) the user is a consumer and the target objects are products being sold through promotional deals.

(d.) the user is an investor and the target objects are publicly traded stocks, mutual funds and/or real estate properties.

(e.) the user is a student and the target objects are classes being offered.

(f.) the user is an activist and the target objects are Congressional bills of potential concern.

(g.) the user is about to send an e-mail message and the target objects are potential recipients who are interested in the content of that message.

(h.) the user is a corporate receptionist receiving incoming e-mail, voice mail or live telephone calls and the target objects are the employees which are the most qualified to handle those incoming media.

(i.) the user is a net-surfer and the target objects are links to pages, servers, or newsgroups available on the World Wide Web which are linked from pages and articles in the on-line newspaper.

(j.) the user is a philanthropist and the target objects are charities.

(k.) the user is ill and the target objects are ads for medical specialists.

(l.) the user is an employee and the target objects are classifieds for potential employers.

(m.) the user is an employer and the target objects are classifieds for potential employees.

(n.) the user is a lonely heart and the target objects are classifieds for potential conversation partners.

(o.) the user is in search of an expert and the target objects are users, with known retrieval habits, of an document retrieval system.

(p.) the user is in need of insurance and the target objects are classifieds for insurance policy offers.

In all these cases, the user wishes to locate some small subset of the target objects—such as the target objects that the user most desires to rent, buy, investigate, meet, read, give mammograms to, insure, and so forth. The task is to help the user identify the most interesting target objects, where the user's interest in a target object is defined to be a numerical measurement of the user's relative desire to locate that object rather than others.

The generality of this problem motivates a general approach to solving the information retrieval problems noted above. It is assumed that many target objects are known to the system for customized electronic identification of desirable objects, and that specifically, the system stores (or has the ability to reconstruct) several pieces of information about each target object. These pieces of information are termed "attributes":

collectively, they are said to form a profile of the target object, or a "target profile." For example, where the system for customized electronic identification of desirable objects is activated to identify selections of interest in a particular category of on-line products for review or purchase by the user, it can be appreciated that there are certain unique sets of attributes which are pertinent to the particular product category of choice. For the application as part of a movie critic column (where the system identifies novel titles and reviews which are most interesting to the user) the system is likely be concerned with the values of attributes such as these:

(a.) title of movie,

(b.) name of director,

(c.) Motion Picture Association of America (MPAA) child-appropriateness rating (0=G, 1=PG, . . .),

(d.) date of release,

(e.) number of stars granted by a particular critic,

(f.) number of stars granted by a second critic,

(g.) number of stars granted by a third critic,

For example, a customized financial news column may be presented to the user in the form of articles which are of

interest to the user. In this case, however, an accordingly those stocks which are most interesting to the user may be presented as well.

(h.). full text of review by the third critic,

(i.). list of customers who have previously rented this movie,

(j.) list of actors.

Each movie has a different set of values for these attributes. This example conveniently illustrates three kinds of attributes. Attributes c–g are numeric attributes, of the sort that might be found in a database record. It is evident that they can be used to help the user identify target objects (movies) of interest. For example, the user might previously have rented many Parental Guidance (PG) films, and many films made in the 1970's. This generalization is useful: new films with values for one or both attributes that are numerically similar to these (such as MPAA rating of 1, release date of 1975) are judged similar to the films the user already likes, and therefore of probable interest. Attributes a–b and h are textual attributes. They too are important for helping the user locate desired films. For example, perhaps the user has shown a past interest in films whose review text (attribute h) contains words like "chase," "explosion," "explosions," "hero," "gripping," and "superb." This generalization is again useful in identifying new films of interest. Attribute i is an associative attribute. It records associations between the target objects in this domain, namely movies, and ancillary target objects of an entirely different sort, namely humans. A good indication that the user wants to rent a particular movie is that the user has previously rented other movies with similar attribute values, and this holds for attribute I just as it does for attributes a–h. For example, if the user has often liked movies that customer C_{17} and customer C_{190} have rented, then the user may like other such movies, which have similar values for attribute i. Attribute j is another example of an associative attribute, recording associations between target objects and actors. Notice that any of these attributes can be made subject to authentication when the profile is constructed, through the use of digital signatures; for example, the target object could be accompanied by a digitally signed note from the MPAA, which note names the target object and specifies its authentic value for attribute c.

These three kinds of attributes are common: numeric, textual, and associative. In the classical information retrieval problem, where the target objects are documents (or more generally, coherent document sections extracted by a text segmentation method), the system might only consider a single, textual attribute when measuring similarity: the full text of the target object. However, a more sophisticated system would consider a longer target profile, including numeric and associative attributes:

(a.) full text of document (textual),

(b.) title (textual),

(c.) author (textual),

(d.) language in which document is written (textual),

(e.) date of creation (numeric),

(f.) date of last update (numeric),

(g.) length in words (numeric),

(h.) reading level (numeric),

(i.) quality of document as rated by a third party editorial agency (numeric),

(j.) list of other readers who have retrieved this document (associative).

As another domain example, consider a domain where the user is an advertiser and the target objects are potential customers. The system might store the following attributes for each target object (potential customer):

(a.) first two digits of zip code (textual),

(b.) first three digits of zip code (textual),

(c.) entire five-digit zip code (textual),

(d.) distance of residence from advertiser's nearest physical storefront (numeric),

(e.) annual family income (numeric),

(f.) number of children (numeric),

(g.) list of previous items purchased by this potential customer (associative),

(h.) list of filenames stored on this potential customer's client computer (associative),

(i.) list of movies rented by this potential customer (associative),

(j.) list of investments in this potential customer's investment portfolio (associative),

(k.) list of documents retrieved by this potential customer (associative),

(l.) written response to Rorschach inkblot test (textual),

(m.) multiple-choice responses by this customer to 20 self-image questions (20 textual attributes).

As always, the notion is that similar consumers buy similar products. It should be noted that diverse sorts of information are being used here to characterize consumers, from their consumption patterns to their literary taste s and psychological peculiarities, and that this fact illustrates both the flexibility and power of the system for customized electronic identification of desirable objects of the present invention. Diverse sorts of information can be used as attributes in other domains as well (as when physical, economic, psychological and interest-related questions are used to profile the applicants to a dating service, which is indeed a possible domain for the present system), and the advertiser domain is simply an example.

As a final domain example, consider a domain where the user is an stock market investor and the target objects are publicly traded corporations. A great many attributes might be used to characterize each corporation, including but not limited to the following:

(a.) type of business (textual),

(b.) corporate mission statement (textual),

(c.) number of employees during each of the last 10 years (ten separate numeric attributes),

(d.) percentage growth in number of employees during each of the last 10 years,

(e.) dividend payment issued in each of the last 40 quarters, as a percentage of current share price,

(f.) percentage appreciation of stock value during each of the last 40 quarters, list of shareholders (associative),

(g.) composite text of recent articles about the corporation in the financial press (textual).

For example, a customized financial news column may be presented to the user in the form of articles which are of interest to the user. In addition, those stocks which are most interesting to the user may be presented as well.

It is worth noting some additional attributes that are of interest in some domains. In the case of documents and certain other domains, it is useful to know the source of each target object (for example, refereed journal article vs. UPI newswire article vs. Usenet newsgroup posting vs. question-answer pair from a question-and-answer list vs. tabloid newspaper article vs. . . .); the source may be represented

as a single-term textual attribute. Important associative attributes for a hypertext document are the list of documents that it links to, and the list of documents that link to it. Documents with similar citations are similar with respect to the former attribute, and documents that are cited in the same places are similar with respect to the latter. A convention may optionally be adopted that any document also links to itself. Especially in systems where users can choose whether or not to retrieve a target object, a target object's popularity (or circulation) can be usefully measured as a numeric attribute specifying the number of users who have retrieved that object. Related measurable numeric attributes that also indicate a kind of popularity include the number of replies to a target object, in the domain where target objects are messages posted to an electronic community such as an computer bulletin board or newsgroup, and the number of links leading to a target object, in the domain where target objects are interlinked hypertext documents on the World Wide Web or a similar system. A target object may also receive explicit numeric evaluations (another kind of numeric attribute) from various groups, such as the Motion Picture Association of America (MPAA), as above, which rates movies' appropriateness for children, or the American Medical Association, which might rate the accuracy and novelty of medical research papers, or a random survey sample of users (chosen from all users or a selected set of experts), who could be asked to rate nearly anything. Certain other types of evaluation, which also yield numeric attributes, may be carried out mechanically. For example, the difficulty of reading a text can be assessed by standard procedures that count word and sentence lengths, while the vulgarity of a text could be defined as (say) the number of vulgar words it contains, and the expertise of a text could be crudely assessed by counting the number of similar texts its author had previously retrieved and read using the invention, perhaps confining this count to texts that have high approval ratings from critics. Finally, it is possible to synthesize certain textual attributes mechanically, for example to reconstruct the script of a movie by applying speech recognition techniques to its soundtrack or by applying optical character recognition techniques to its closed-caption subtitles.

Decomposing Complex Attributes

Although textual and associative attributes are large and complex pieces of data, for information retrieval purposes they can be decomposed into smaller, simpler numeric attributes. This means that any set of attributes can be replaced by a (usually larger) set of numeric attributes, and hence that any profile can be represented as a vector of numbers denoting the values of these numeric attributes. In particular, a textual attribute, such as the full text of a movie review, can be replaced by a collection of numeric attributes that represent scores to denote the presence and significance of the words "aardvark," "aback," "abacus," and so on through "zymurgy" in that text. The score of a word in a text may be defined in numerous ways. The simplest definition is that the score is the rate of the word in the text, which is computed by computing the number of times the word occurs in the text, an d dividing this number by the total number of words in the text. This sort of score is often called the "term frequency" (TF) of the word. The definition of term frequency may optionally be modified to weight different portions of the text unequally: for example, any occurrence of a word in the text's title might be counted as a 3-fold or more generally k-fold occurrence (as if the title had been repeated k times within the text), in order to reflect a heuristic assumption that the words in the title are particularly important indicators of the text's content or topic.

However, for lengthy textual attributes, such as the text of an entire document, the score of a word is typically defined to be not merely its term frequency, but its term frequency multiplied by the negated logarithm of the word's "global frequency," as measured with respect to the textual attribute in question. The global frequency of a word, which effectively measures the word's uninformativeness, is a fraction between 0 and 1, defined to be the fraction of all target objects for which the textual attribute in question contains this word. This adjusted score is often known in the art as TF/IDF ("term frequency times inverse document frequency"). When global frequency of a word is taken into account in this way, the common, uninformative words have scores comparatively close to zero, no matter how often or rarely they appear in the text. Thus, their rate has little influence on the object's target profile. Alternative methods of calculating word scores include latent semantic indexing or probabilistic models.

Instead of breaking the text into its component words, one could alternatively break the text into overlapping word bigrams (sequences of 2 adjacent words), or more generally, word n-grams. These word n-grams may be scored in the same way as individual words. Another possibility is to use character n-grams. For example, this sentence contains a sequence of overlapping character 5-grams which starts "for e", "or ex", "r exa", "exam", "examp", etc. The sentence may be characterized, imprecisely but usefully, by the score of each possible character 5-gram ("aaaaa", "aaaab", . . . "zzzzz") in the sentence. Conceptually speaking, in the character 5-gram case, the textual attribute would be decomposed into at least 26^5=11,881,376 numeric attributes. Of course, for a given target object, most of these numeric attributes have values of 0, since most 5-grams do not appear in the target object attributes. These zero values need not be stored anywhere. For purposes of digital storage, the value of a textual attribute could be characterized by storing the set of character 5-grams that actually do appear in the text, together with the nonzero score of each one. Any 5-gram that is no t included in the set can be assumed to have a score of zero. The decomposition of textual attributes is not limited to attributes whose values are expected to be long texts. A simple, one-term textual attribute can be replaced by a collection of numeric attributes in exactly the same way. Consider again the case where the target objects are movies. The "name of director" attribute, which is textual, can be replaced by numeric attributes giving the scores for "Federico-Fellini," "Woody-Allen," "Terence-Davies," and so forth, in that attribute. For these one-term textual attributes, the score of a word is usually defined to be its rate in the text, without any consideration of global frequency. Note that under these conditions, one of the scores is 1, while the other scores are 0 and need not be stored. For example, if Davies did direct the film, then it is "Terence-Davies" whose score is 1, since "Terence-Davies" constitutes 100% of the words in the textual value of the "name of director" attribute. It might seem that nothing has been gained over simply regarding the textual attribute as having the string value "Terence-Davies." However, the trick of decomposing every non-numeric attribute into a collection of numeric attributes proves useful for the clustering and decision tree methods described later, which require the attribute values of different objects to be averaged and/or ordinally ranked. Only numeric attributes can be averaged or ranked in this way. Just as a textual attribute may be decomposed into a number of component terms (letter or word n-grams), an associative attribute may be decomposed into a number of component associations. For instance, in a

domain where the target objects are movies, a typical associative attribute used in profiling a movie would be a list of customers who have rented that movie. This list can be replaced by a collection of numeric attributes, which give the "association scores" between the movie and each of the customers known to the system. For example, the 165th such numeric attribute would be the association score between the movie and customer #165, where the association score is defined to be 1 if customer #165 has previously rented the movie, and 0 otherwise. In a subtler refinement, this association score could be defined to be the degree of interest, possibly zero, that customer #165 exhibited in the movie, as determined by relevance feedback (as described below). As another example, in a domain where target objects are companies, an associative attribute indicating the major shareholders of the company would be decomposed into a collection of association scores, each of which would indicate the percentage of the company (possibly zero) owned by some particular individual or corporate body. Just as with the term scores used in decomposing lengthy textual attributes, each association score may optionally be adjusted by a multiplicative factor: for example, the association score between a movie and customer #165 might be multiplied by the negated logarithm of the "global frequency" of customer #165, i.e., the fraction of all movies that have been rented by customer #165. Just as with the term scores used in decomposing textual attributes, most association scores found when decomposing a particular value of an associative attribute are zero, and a similar economy of storage may be gained in exactly the same manner by storing a list of only those ancillary objects with which the target object has a nonzero association score, together with their respective association scores.

Similarity Measures

What does it mean for two target objects to be similar? More precisely, how should one measure the degree of similarity? Many approaches are possible and any reasonable metric that can be computed over the set of target object profiles can be used, where target objects are considered to be similar if the distance between their profiles is small according to this metric. Thus, the following preferred embodiment of a target object similarity measurement system has m any variations.

First, define the distance between two values of a given attribute according to whether the attribute is a numeric, associative, or textual attribute. If the attribute is numeric, then the distance between two values of the attribute is the absolute value of the difference between the two values. (Other definitions are also possible: for example, the distance between prices p1 and p2 might be defined by $|(p1-p2)|/(\max(p1,p2)+1)$, to recognize that when it comes to customer interest, $5000 and $5020 are very similar, whereas $3 and $23 are not.) If the attribute is associative, then its value V may be decomposed as described above into a collection of real numbers, representing the association scores between the target object in question and various ancillary objects. V may therefore be regarded as a vector with components V_1, V_2, V_3, etc., representing the association scores between the object and ancillary objects 1, 2, 3, etc., respectively. The distance between two vector values V and U of an associative attribute is then computed using the angle distance measure, $\arccos(VU^t/\mathrm{sqrt}((Vv^t)(UU^t))$. (Note that the three inner products in this expression have the form $XY^t=X_1Y_1+X_2Y_2+X_3Y_3+\ldots$, and that for efficient computation, terms of the form $X_i Y_i$ may be omitted from this sum if either of the scores X_i and Y_i is zero.) Finally, if the attribute is textual, then its value V may be decomposed as described above into a collection of real numbers, representing the scores of various word n-grams or character n-grams in the text. Then the value V may again be regarded as a vector, and the distance between two values is again defined via the angle distance measure. Other similarity metrics between two vectors, such as the dice measure, may be used instead. It happens that the obvious alternative metric, Euclidean distance, does not work well: even similar texts tend not to overlap substantially in the content words they use, so that texts encountered in practice are all substantially orthogonal to each other, assuming that TF/IDF scores are used to reduce the influence of non-content words. The scores of two words in a textual attribute vector may be correlated; for example, "Kennedy" and "JFK" tend to appear in the same documents. Thus it may be advisable to alter the text somewhat before computing the scores of terms in the text, by using a synonym dictionary that groups together similar words. The effect of this optional prealteration is that two texts using related words are measured to be as similar as if they had actually used the same words. One technique is to augment the set of words actually found in the article with a set of synonyms or other words which tend to co-occur with the words in the article, so that "Kennedy" could be added to every article that mentions "JFK." Alternatively, words found in the article may be wholly replaced by synonyms, so that "JFK" might be replaced by "Kennedy" or by "John F. Kennedy" wherever it appears. In either case, the result is that documents about Kennedy and documents about JFK are adjudged similar. The synonym dictionary may be sensitive to the topic of the document as a whole; for example, it may recognize that "crane" is likely to have a different synonym in a document that mentions birds than in a document that mentions construction. A related technique is to replace each word by its morphological stem, so that "staple", "stapler", and "staples" are all replaced by "staple." Common function words ("a", "and", "the" . . .) c an influence the calculated similarity of texts without regard to their topics, and so are typically removed from the text before the scores of terms in the text are computed. A more general approach to recognizing synonyms is to use a revised measure of the distance between textual attribute vectors V and U, namely $\arccos(AV(AU)^t/\mathrm{sqrt}(AV(AV)^t\,AU(AU)^t)$, where the matrix A is the dimensionality-reducing linear transformation (or an approximation thereto) determined by collecting the vector values of the textual attribute, for all target objects known to the system, and applying singular value decomposition to the resulting collection. The same approach can be applied to the vector values of associative attributes. The above definitions allow us to determine how close together two target objects are with respect to a single attribute, whether numeric, associative, or textual. The distance between two target objects X and Y with respect to their entire multi-attribute profiles P_X and P_Y is then denoted $d(X,Y)$ or $d(P_X, P_Y)$ and defined as:

$$(((\text{distance with respect to attribute a})(\text{weight of attribute a}))^k+((\text{distance with respect to attribute b})(\text{weight of attribute b}))^k+((\text{distance with respect to attribute c})(\text{weight of attribute c}))^k+\ldots)^k$$

where k is a fixed positive real number, typically 2, and the weights are non-negative real numbers indicating the relative importance of the various attributes. For example, if the target objects are consumer goods, and the weight of the "color" attribute is comparatively very small, then price is not a consideration in determining similarity: a user who likes a brown massage cushion is predicted to show equal interest in the same cushion manufactured in blue, and

vice-versa. On the other hand, if the weight of the "color" attribute is comparatively very high, then users are predicted to show interest primarily in products whose colors they have liked in the past: a brown massage cushion and a blue massage cushion are not at all the same kind of target object, however similar in other attributes, and a good experience with one does not by itself inspire much interest in the other. Target objects may be of various sorts, and it is sometimes advantageous to use a single system that is able to compare tar get objects of distinct sorts. For example, in a system where some target objects are novels while other target objects are movies, it is desirable to judge a novel and a movie similar if their profiles show that similar users like them (an associative attribute). However, it is important to note that certain attributes specified in the movie's target profile are undefined in the novel's target profile, and vice versa: a novel has no "cast list" associative attribute and a movie has no "reading level" numeric attribute. In general, a system in which target objects fall into distinct sorts may sometimes have to measure the similarity of two target objects for which somewhat different sets of attributes are defined. This requires an extension to the distance metric d(*,*) defined above. In certain applications, it is sufficient when carrying out such a comparison simply to disregard attributes that are not defined for both target objects: this allows a cluster of novels to be matched with the most similar cluster of movies, for example, by considering only those attributes that novels and movies have in common. However, while this method allows comparisons between (say) novels and movies, it does not define a proper metric over the combined space of novel s and movies and therefore does not allow clustering to be applied to the set of all target objects. When necessary for clustering or other purposes, a metric that allows comparison of any two target objects (whether of the same or different sorts) can be defined as follows. If a is an attribute, then let Max(a) be an upper bound on the distance between two values of attribute a; notice that if attribute a is an associative or textual attribute, this distance is an angle determined by arccos, so that Max(a) may be chosen to be 180 degrees, while if attribute a is a numeric attribute, a sufficiently large number must be selected by the system designers. The distance between two values of attribute a is given as before in the case where both values are defined; the distance between two undefined values is taken to be zero; finally, the distance between a defined value and an undefined value is always taken to be Max(a)/2. This allows us to determine how close together two target objects are with respect to an attribute a, even if attribute a does not have a defined value for both target objects. The distance d(*,*) between two target objects with respect to their entire multi-attribute profiles is then given in terms of these individual attribute distances exactly as before. It is assumed that one attribute in such a system specifies the sort of target object ("movie", "novel", etc.), and that this attribute may be highly weighted if target objects of different sorts are considered to be very different despite any attributes they may have in common.

UTILIZING THE SIMILARITY MEASUREMENT

Matching Buyers and Sellers

A simple application of the similarity measurement is a system to match buyers with sellers in small-volume markets, such as used cars and other used goods, artwork, or employment. Sellers submit profiles of the goods (target objects) they want to sell, and buyers submit profiles of the goods (target objects) they want to buy. Participants may submit or withdraw these profiles at any time. The system for customized electronic identification of desirable objects computes the similarities between seller-submitted profiles and buyer-submitted profiles, and when two profiles match closely (i.e., the similarity is above a threshold), the corresponding seller and buyer are notified of each other's identities. To prevent users from being flooded with responses, it may be desirable to limit the number of notifications each user receives to a fixed number, such as ten per day.

Filtering: Relevance Feedback

A filtering system is a device that can search through many target objects and estimate a given user's interest in each target object, so as to identify those that are of greatest interest to the user. The filtering system uses relevance feed back to refine its knowledge of the user's interests: whenever the filtering system identifies a target object as potentially interesting to a user, the user (if an on-line user) provides feedback as to whether or not that target object really is of interest. Such feedback is stored long-term in summarized form, as part of a database of user feedback information, and may be provided either actively or passively. In active feedback, the user explicitly indicates his or her interest, for instance, on a scale of −2 (active distaste) through 0 (no special interest) to 10 (great interest). In passive feedback, the system infers the user's interest from the user's behavior. For example, if target objects are textual documents, the system might monitor which documents the user chooses to read, or not to read, and how much time the user spends reading them. A typical formula for assessing interest in a document via passive feedback, in this domain, on a scale of 0 to 10, might be:

- +2 if the second page is viewed,
- +2 if all pages are viewed,
- +2 if more than 30 seconds was spent viewing the document,
- +2 if more than one minute was spent viewing the document,
- +2 if the minutes spent viewing the document are greater than half the number of pages.

If the target objects are electronic mail messages, interest points might also be added in the case of a particularly lengthy or particularly prompt reply. If the target objects are purchasable goods, interest points might be added for target objects that the user actually purchases, with further points in the case of a large-quantity or high-price purchase. In any domain, further points might be added for target objects that the user accesses early in a session, on the grounds that users access the objects that most interest them first. Other potential sources of passive feedback include an electronic measurement of the extent to which the user's pupils dilate while the user views the target object or a description of the target object. It is possible to combine active and passive feedback. One option is to take a weighted average of the two ratings. Another option is to use passive feedback by default, but to allow the user to examine and actively modify the passive feedback score. In the scenario above, for instance, an uninteresting article may sometimes remain on the display device for a long period while the user is engaged in unrelated business; the passive feedback score is then inappropriately high, and the user may wish to correct it before continuing. In the preferred embodiment of the invention, a visual indicator, such as a sliding bar or indicator needle on the user's screen, can be used to continuously display the passive feedback score estimated by the system for the target object being viewed, unless the user has manually adjusted the indicator by a mouse operation or other means in order to reflect a different score for this target object, after which

the indicator displays the active feedback score selected by the user, and this active feedback score is used by the system instead of the passive feedback score. In a variation, the user cannot see or adjust the indicator until just after the user has finished viewing the target object. Regardless how a user's feedback is computed, it is stored long-term as part of that user's target profile interest summary.

Filtering: Determining Topical Interest Through Similarity

Relevance feedback only determines the user's interest in certain target objects: namely, the target objects that the user has actually had the opportunity to evaluate (whether actively or passively). For target objects that the user has not yet seen, the filtering system must estimate the user's interest. This estimation task is the heart of the filtering problem, and the reason that the similarity measurement is important. More concretely, the preferred embodiment of the filtering system is a news clipping service that periodically presents the user with news articles of potential interest. The user provides active and/or passive feedback to the system relating to these presented articles. However, the system does not have feedback information from the user for articles that have never been presented to the user, such as new articles that have just been added to the database, or old articles that the system chose not to present to the user. Similarly, in the dating service domain where target objects are prospective romantic partners, the system has only received feedback on old flames, not on prospective new loves.

As shown in flow diagram form in FIG. **12**, the evaluation of the likelihood of interest in a particular target object for a specific user can automatically be computed. The interest that a given target object X holds for a user U is assumed to be a sum of two quantities: q(U, X), the intrinsic "quality" of X, plus f(U, X), the "topical interest" that users like U have in target objects like X. For any target object X, the intrinsic quality measure q(U, X) is easily estimated at steps **1201–1203** directly from numeric attributes of the target object X. The computation process begins at step **1201**, where certain designated numeric attributes of target object X are specifically selected, which attributes by their very nature should be positively or negatively correlated with users' interest. Such attributes, termed "quality attributes," have the normative property that the higher (or in some cases lower) their value, the more interesting a user is expected to find them. Quality attributes of target object X may include, but are not limited to, target object X's popularity among users in general, the rating a particular reviewer has given target object X, the age (time since authorship—also known as outdatedness) of target object X, the number of vulgar words used in target object X, the price of target object X, and the amount of money that the company selling target object X has donated to the user's favorite charity. At step **1202**, each of the selected attributes is multiplied by a positive or negative weight indicative of the strength of user U's preference for those target objects that have high values for this attribute, which weight must be retrieved from a data file storing quality attribute weights for the selected user. At step **1203**, a weighted sum of the identified weighted selected attributes is computed to determine the intrinsic quality measure q(U, X). At step **1204**, the summarized weighted relevance feedback data is retrieved, wherein some relevance feedback points are weighted more heavily than others and the stored relevance data can be summarized to some degree, for example by the use of search profile sets. The more difficult part of determining user U's interest in target object X is to find or compute at step **1205** the value of f(U, X), which denotes the topical interest that users like U generally have in target objects like X. The method of determining a user's interest relies on the following heuristic: when X and Y are similar target objects (have similar attributes), and U and V are similar users (have similar attributes), then topical interest f(U, X) is predicted to have a similar value to the value of topical interest f(V, Y). This heuristic leads to an effective method because estimated values of the topical interest function f(*, *) are actually know n for certain arguments to that function: specifically, if user V has provided a relevance-feedback rating of r(V, Y) for target object Y, then insofar as that rating represents user V's true interest in target object Y, we have r(V, Y)=q(V, Y)+f(V, Y) and can estimate f(V, Y) as r(V, Y)−q(V, Y). Thus, the problem of estimating topical interest at all points becomes a problem of interpolating among these estimates of topical interest at selected points, such as the feedback estimate of f(V, Y) a s r(V, Y)−q(V, Y). This interpolation can be accomplished with any standard smoothing technique, using as input the known point estimates of the value of the topical interest function f(*, *), and determining as output a function that approximates the entire topical interest function f(*, *).

Not all point estimates of the topical interest function f(*, *) should be given equal weight as inputs to the smoothing algorithm. Since passive relevance feedback is less reliable than active relevance feedback, point estimates made from passive relevance feedback should be weighted less heavily than point estimates made from active relevance feedback, or even not used at all. In most domains, a user's interests may change over time and, therefore, estimates of topic al interest that derive from more recent feedback should also be weighted more heavily. A user's interests may vary according to mood, so estimates of topical interest that derive from the current session should be weighted more heavily for the duration of the current session, and past estimates of topical interest made at approximately the current time of day or on the current weekday should be weighted more heavily. Finally, in domains where users are trying to locate target objects of long-term interest (investments, romantic partners, pen pals, employers, employees, suppliers, service providers) from the possibly meager information provided by the target profiles, the users are usually not in a position to provide reliable immediate feedback on a target object, but can provide reliable feedback at a later date. An estimate of topical interest f(V, Y) should be weighted more heavily if user V has had more experience with target object Y. Indeed, a useful strategy is for the system to track long-term feedback for such target objects. For example, if target profile Y was created in 1990 to describe a particular investment that was available in 1990, and that was purchased in 1990 by user V, then the system solicits relevance feedback from user V in the years 1990, 1991, 1992, 1993, 1994, 1995, etc., and treats these as successively stronger indications of user V's true interest in target profile Y, and thus as indications of user V's likely interest in new investments whose current profiles resemble the original 1990 investment profile Y. In particular, if in 1994 and 1995 user V is well-disposed toward his or her 1990 purchase of the investment described by target profile Y, then in those years and later, the system tends to recommend additional investments when they have profiles like target profile Y, on the grounds that they too will turn out to be satisfactory in 4 to 5 years. It makes these recommendations both to user V and to users whose investment portfolios and other attributes are similar to user V's. The relevance feedback provided by user V in this case may be either active (feedback=satisfaction ratings provided by the

investor V) or passive (feedback=difference between average annual return of the investment and average annual return of the Dow Jones index portfolio since purchase of the investment, for example).

To effectively apply the smoothing technique, it is necessary to have a definition of the similarity distance between (U, X) and (V, Y), for any users U and V and any target objects X and Y. We have already seen how to define the distance d(Y, Y) between two target objects X and Y, given their attributes. We may regard a pair such as (U, X) as an extended object that bears all the attributes of target X and all the attributes of user U; then the distance between (U, X) and (V, Y) may be computed in exactly the same way. This approach requires user U, user V, and all other users to have some attributes of their own stored in the system: for example, age (numeric), social security number (textual), and list of documents previously retrieved (associative). It is these attributes that determine the notion of "similar users." Thus it is desirable to generate profiles of users (termed "user profiles") as well as profiles of target objects (termed "target profiles"). Some attributes employed for profiling users may be related to the attributes employed for profiling target objects: for example, using associative attributes, it is possible to characterize target objects such as X by the interest that various users have shown in them, and simultaneously to characterize users such as U by the interest that they have shown in various target objects. In addition, user profiles may make use of any attributes that are useful in characterizing humans, such as those suggested in the example domain above where target objects are potential consumers. Notice that user U's interest can be estimated even if user U is a new user or an off-line user who has never provided any feedback, because the relevance feedback of users whose attributes are similar to U's attributes is taken into account.

For some uses of filtering systems, when estimating topical interest, it is appropriate to make an additional "presumption of no topical interest" (or "bias toward zero"). To understand the usefulness of such a presumption, suppose the system needs to determine whether target object X is topically interesting to the user U, but that users like user U have never provided feedback on target objects even remotely like target object X. The presumption of no topical interest says that if this is so, it is because users like user U are simply not interested in such target objects and therefore do not seek them out and interact with them. On this presumption, the system should estimate topical interest f(U, X) to be low. Formally, this example has the characteristic that (U, X) is far away from all the points (V, Y) where feedback is available. In such a case, topical interest f(U, X) is presumed to be close to zero, even if the value of the topical interest function f(*, *) is high at all the faraway surrounding points at which its value is known. When a smoothing technique is used, such a presumption of no topical interest can be introduced, if appropriate, by manipulating the input to the smoothing technique. In addition to using observed values of the topical interest function f(*, *) as input, the trick is to also introduce fake observations of the form topical interest f(V, Y)=0 for a lattice of points (V, Y) distributed throughout the multidimensional space. These fake observations should be given relatively low weight as inputs to the smoothing algorithm. The more strongly they are weighted, the stronger the presumption of no interest.

The following provides another simple example of an estimation technique that has a presumption of no interest. Let g be a decreasing function from non-negative real numbers to non-negative real numbers, such as $g(x)=e^{x}$ or $g(x)=\min(1, x^{-k})$ where k>1. Estimate topical interest f(U, X) with the following g-weighted average:

$$f(U, X) = \frac{\Sigma((r(V, Y) - q(V, Y)) * g(\text{distance } \Phi(U, X)\Lambda(V, Y)}{\Sigma g(\text{distance } \Phi(U, V)\Lambda(V, Y))}$$

Here the summations are over all pairs (V, Y) such that user V has provided feedback r(V, Y) on target object Y, i.e., all pairs (V, Y) such that relevance feedback r(V, Y) is defined. Note that both with this technique and with conventional smoothing techniques, the estimate of the topical interest f(U, X) is not necessarily equal to r(U, X)−q(U, X), even when r(U, X) is defined.

Filtering: Adjusting Weights and Residue Feedback

The method described above requires the filtering system to measure distances between (user, target object) pairs, such as the distance between (U, X) and (V, Y). Given the means described earlier for measuring the distance between two multi-attribute profiles, the method must therefore associate a weight with each attribute used in the profile of (user, target object) pairs, that is, with each attribute used to profile either users or target objects. These weights specify the relative importance of the attributes in establishing similarity or difference, and therefore, in determining how topical interest is generalized from one (user, target object) pair to another. Additional weights determine which attributes of a target object contribute to the quality function q, and by how much.

It is possible and often desirable for a filtering system to store a different set of weights for each user. For example, a user who thinks of two-star films as having materially different topic and style from four-star films wants to assign a high weight to "number of stars" for purposes of the similarity distance measure d(*, *); this means that interest in a two-star film does not necessarily signal interest in an otherwise similar four-star film, or vice-versa. If the user also agrees with the critics, and actually prefers four-star films, the user also wants to assign "number of stars" a high positive weight in the determination of the quality function q. In the same way, a user who dislikes vulgarity wants to assign the "vulgarity score" attribute a high negative weight in the determination of the quality function q, although the "vulgarity score" attribute does not necessarily have a high weight in determining the topical similarity of two films.

Attribute weights (of both sorts) may be set or adjusted by the system administrator or the individual user, on either a temporary or a permanent basis.

However, it is often desirable for the filtering system to learn attribute weights automatically, based on relevance feedback. The optimal attribute weights for a user U are those that allow the most accurate prediction of user U's interests. That is, with the distance measure and quality function defined by these attribute weights, user U's interest in target object X, q(U, X)+f(U, X), can be accurately estimated by the techniques above. The effectiveness of a particular set of attribute weights for user U can therefore be gauged by seeing how well it predicts user U's known interests.

Formally, suppose that user U has previously provided feedback on target objects $X_1, X_2, X_3, \ldots X_n$, and that the feedback ratings are $r(U, X_1)$, $r(U, X_2)$, $r(U, X_3)$, . . . $r(U, X_n)$. Values of feedback ratings r(*,*) for other users and other target objects may also be known. The system may use the following procedure to gauge the effectiveness of the set of attribute weights it currently stores for user U: (I) For each 1<=I<=n, use the estimation techniques to estimate $q(U, X_i)+f(U, X_i)$ from all known values of feedback ratings

r. Call this estimate a_i. (ii) Repeat step (i), but this time make the estimate for each 1<=i<=n without using the feedback ratings r(U, X_j) as input, for any j such that the distance d(X_i, X_j) is smaller than a fixed threshold. That is, estimate each q(U, X_i)+f(U, X_i) from other values of feedback rating r only; in particular, do not use r(U, X_i) itself Call this estimate b_i. The difference a_i–b_i is herein termed the "residue feedback r_{res}(U, X_i) of user U on target object X_i." (iii) Compute user U's error measure, $(a_1-b_1)^2+(a_2-b_2)^2+(a_3-b_3)^2+ \ldots +(a_n-b_n)^2$.

A gradient-descent or other numerical optimization method may be used to adjust user U's attribute weights so that this error measure reaches a (local) minimum. This approach tends to work best if the smoothing technique used in estimation is such that the value of f(V, Y) is strongly affected by the point estimate r(V, Y)–q(V, Y) when the latter value is provided as input. Otherwise, the presence or absence of the single input feedback rating r(U, X_i), in steps (i)–(ii) may not make a_i and b_i very different from each other. A slight variation of this learning technique adjusts a single global set of at tribute weights for all users, by adjusting the weights so as to minimize not a particular user's error measure but rather the total error measure of all users. These global weights are used as a default initial setting for a new user who has not yet provided any feedback. Gradient descent can then be employed to adjust this user's individual weights over time. Even when the attribute weights are chosen to minimize the error measure for user U, the error measure is generally still positive, meaning that residue feedback from user U has not been reduced to 0 on all target objects. It is useful to note that high residue feedback from a user U on a target object X indicates that user U liked target object X unexpectedly well given its profile, that is, better than the smoothing model could predict from user U's opinions on target objects with similar profiles. Similarly, low residue feedback indicates that user U liked target object X less than was expected. By definition, this unexplained preference or dispreference cannot be the result of topical similarity, and therefore must be regarded as an indication of the intrinsic quality of target object X. It follows that a useful quality attribute for a target object X is the average amount of residue feedback r_{res}(V, X) from users on that target object, averaged over all users V who have provided relevance feedback on the target object. In a variation of this idea, residue feedback is never averaged indiscriminately over all users to form a new attribute, but instead is smoothed to consider users' similarity to each other. Recall that the quality measure q(U, X) depends on the user U as well as the target object X, so that a given target object X may be perceived by different users to have different quality. In this variation, as before, q(U, X) is calculated as a weighted sum of various quality attributes that are dependent only on X, but then an additional term is added, namely an estimate of r_{res} (U, X) found by applying a smoothing algorithm to known values of rres (V, X). Here V ranges over all users who have provided relevance feedback on target object X, and the smoothing algorithm is sensitive to the distances d(U, V) from each such user V to user U.

Using the Similarity Computation for Clustering

A method for defining the distance between any pair of target objects was disclosed above. Given this distance measure, it is simple to apply a standard clustering algorithm, such as k-means, to group the target objects into a number of clusters, in such a way that similar target objects tend to be grouped in the same cluster. It is clear that the resulting clusters can be used to improve the efficiency of matching buyers and sellers in the application described in section "Matching Buyers and Sellers" above: it is not necessary to compare every buy profile to every sell profile, but only to compare buy profiles and sell profiles that are similar enough to appear in the same cluster. As explained below, the results of the clustering procedure can also be used to make filtering more efficient, and in the service of querying and browsing tasks.

The k-means clustering method is familiar to those skilled in the art. Briefly put, it finds a grouping of points (target profiles, in this case, whose numeric coordinates are given by numeric decomposition of their attributes as described above) to minimize the distance between points in the clusters and the centers of the clusters in which they are located. This is done by alternating between assigning each point to the cluster which has the nearest center and then, once the points have been assigned, computing the (new) center of each cluster by averaging the coordinates of the points (target profiles) located in this cluster. Other clustering methods can be used, such as "soft" or "fuzzy" k-means clustering, in which objects are allowed to belong to more than one cluster. This can be cast as a clustering problem similar to the k-means problem, but now the criterion being optimized is a little different:

$$\Sigma_i \Sigma_C j_{ic} d(x_i, \bar{x}_c)$$

where C ranges over cluster numbers, i ranges over target objects, x_i is the numeric vector corresponding to the profile of target object number i, _C is the mean of all the numeric vectors corresponding to target profiles of target objects in cluster number C, termed the "cluster profile" of cluster C, d(*, *) is the metric used to measure distance between two target profiles, and i_{iC} is a value between 0 and 1 that indicates how much target object number i is associated with cluster number C, where i is an indicator matrix with the property that for each i, SUM SUB C I SUB iC=1. For k-means clustering, i_{iC} is either 0 or 1.

Any of these basic types of clustering might be used by the system:

1) Association-based clustering, in which profiles contain only associative attributes, and thus distance is defined entirely by associations. This kind of clustering generally (a) clusters target objects based on the similarity of the users who like them or (b) clusters users based on the similarity of the target objects they like. In this approach, the system does not need any information about target objects or users, except for their history of interaction with each other.

2) Content-based clustering, in which profiles contain only non-associative attributes. This kind of clustering (a) clusters target objects based on the similarity of their non-associative attributes (such as word frequencies) or (b) clusters users based on the similarity of their non-associative attributes (such as demographics and psychographics). In this approach, the system does not need to record any information about users' historical patterns of information access, but it does need information about the intrinsic properties of users and/or target objects.

3) Uniform hybrid method, in which profiles may contain both associative and non-associative attributes. This method combines 1a and 2a, or 1b and 2b. The distance d(P_X, P_Y) between two profiles P_X and P_Y may be computed by the general similarity-measurement methods described earlier.

4) Sequential hybrid method. First apply the k-means procedure to do 1a, so that articles are labeled by cluster based on which user read them, then use supervised clustering (maximum likelihood discriminant methods) using the word frequencies to do the process of method 2a described above. This tries to use knowledge of who read what to do a better job of clustering based on word frequencies. One could similarly combine the methods 1b and 2b described above.

Hierarchical clustering of target objects is often useful. Hierarchical clustering produces a tree which divides the target objects first into two large clusters of roughly similar objects; each of these clusters is in turn divided into two or more smaller clusters, which in turn are each divided into yet smaller clusters until the collection of target objects has been entirely divided into "clusters" consisting of a single object each, as diagrammed in FIG. **8** In this diagram, the node d denotes a particular target object d, or equivalently, a single-member cluster consisting of this target object. Target object d is a member of the cluster (a, b, d), which is a subset of the cluster (a, b, c, d, e, f), which in turn is a subset of all target objects. The tree shown in FIG. **8** would be produced from a set of target objects such as those shown geometrically in FIG. **7**. In FIG. **7**, each letter represents a target object, and axes x1 and x2 represent two of the many numeric attributes on which the target objects differ. Such a cluster tree may be created by hand, using human judgment to form clusters and subclusters of similar objects, or may be created automatically in either of two standard ways: top-down or bottom-up. In top-down hierarchical clustering, the set of all target objects in FIG. **7** would be divided into the clusters (a, b, c, d, e, f) and (g, h, i, j, k). The clustering algorithm would then be reapplied to the target objects in each cluster, so that the cluster (g, h, i, j, k) is subpartitioned into the clusters (g, k) and (h, i, j), and so on to arrive at the tree shown in FIG. **8**. In bottom-up hierarchical clustering, the set of all target objects in FIG. **7** would be grouped into numerous small clusters, namely (a, b), d, (c, f), e, (g,k), (h, i), and j. These clusters would then themselves be grouped into the larger clusters (a, b, d), (c, e, f), (g, k), and (h, i, j), according to their cluster profiles. These larger clusters would themselves be grouped into (a, b, c, d, e, f) and (g, k, h, i, j), and so on until all target objects had been grouped together, resulting in the tree of FIG. **8**. Note that for bottom-up clustering to work, it must be possible to apply the clustering algorithm to a set of existing clusters. This requires a notion of the distance between two clusters. The method disclosed above for measuring the distance between target objects can be applied directly, provided that clusters are profiled in the same way as target objects. It is only necessary to adopt the convention that a cluster's profile is the average of the target profiles of all the target objects in the cluster; that is, to determine the cluster's value for a given attribute, take the mean value of that attribute across all the target objects in the cluster. For the mean value to be well-defined, all attributes must be numeric, so it is necessary as usual to replace each textual or associative attribute with its decomposition into numeric attributes (scores), as described earlier. For example, the target profile of a single Woody Allen film would assign "Woody-Allen" a score of 1 in the "name-of-director" field, while giving "Federico-Fellini" and "Terence-Davies" scores of 0. A cluster that consisted of 20 films directed by Allen and 5 directed by Fellim would be profiled with scores of 0.8, 0.2, and 0 respectively, because, for example, 0.8 is the average of 20 ones and 5 zeros.

Searching for Target Objects

Given a target object with target profile P, or alternatively given a search profile P, a hierarchical cluster tree of target objects makes it possible for the system to search efficiently for target objects with target profiles similar to P. It is only necessarily to navigate through the tree, automatically, in search of such target profiles. The system for customized electronic identification of desirable objects begins by considering the largest, top-level clusters, and selects the cluster whose profile is most similar to target profile P. In the event of a near-tie, multiple clusters may be selected. Next, the system considers all subclusters of the selected clusters, and this time selects the subclusters or subclusters whose profiles are closest to target profile P. This refinement process is iterated until the clusters selected on a given step are sufficiently small, and these are the desired clusters of target objects with profiles most similar to target profile P. Any hierarchical cluster tree therefore serves as a decision tree for identifying target objects. In pseudo-code form, this process is as follows (and in flow diagram form in FIGS. **13A** and **13B**):

1. Initialize list of identified target objects to the empty list at step **13A00**
2. Initialize the current tree T to be the hierarchical cluster tree of all objects at step **13A01** and at step **13A02** scan the current cluster tree for target objects similar to P, using the process detailed in FIG. **13B**. At step **13A03**, the list of target objects is returned.
3. At step **13B00**, the variable I is set to 1 and for each child subtree Ti of the root of tree T, is retrieved.
4. At step **13B02**, calculate $d(P, p_i)$, the similarity distance between P and p_i,
5. At step **13B03**, if $d(P, p_i)<t$, a threshold, branch to one of two options
6. If tree Ti contains only one target object at step **13B04**, add that target object to list of identified target objects at step **13B05** and advance to step **13B07**.
7. If tree Ti contains multiple target objects at step **13B04**, scan the ith child subtree for target objects similar to P by invoking the steps of the process of FIG. **13B** recursively and then recurse to step 3 (step **13A01** in FIG. **13A**) with T bound for the duration of the recursion to tree Ti, in order to search in tree Ti for target objects with profiles similar to P.

In step 5 of this pseudo-code, smaller thresholds are typically used at lower levels of the tree, for example by making the threshold an affine function or other function of the cluster variance or cluster diameter of the cluster p_i. If the cluster tree is distributed across a plurality of servers, as described in the section of this description titled "Network Context of the Browsing System", this process may be executed in distributed fashion as follows: steps 3–7 are executed by the server that stores the root node of hierarchical cluster tree T, and the recursion in step 7 to a subcluster tree T_i involves the transmission of a search request to the server that stores the root node of tree T_i, which server carries out the recursive step upon receipt of this request. Steps 1–2 are carried out by the processor that initiates the search, and the server that executes step 6 must send a message identifying the target object to this initiating processor, which adds it to the list.

Assuming that low-level clusters have been already been formed through clustering, there are alternative search methods for identifying the low-level cluster whose profile is most similar to a given target profile P. A standard back-propagation neural net is one such method: it should be

trained to take the attributes of a target object as input, and produce as output a unique pattern that can be used to identify the appropriate low-level cluster. For maximum accuracy, low-level clusters that are similar to each other (close together in the cluster tree) should be given similar identifying patterns. Another approach is a standard decision tree that considers the attributes of target profile P one at a time until it can identify the appropriate cluster. If profiles are large, this may be more rapid than considering all attributes. A hybrid approach to searching uses distance measurements as described above to navigate through the top few levels of the hierarchical cluster tree, until it reaches an cluster of intermediate size whose profile is similar to target profile P, and then continues by using a decision tree specialized to search for low-level subclusters of that intermediate cluster.

One use of these searching techniques is to search for target objects that match a search profile from a user's search profile set. This form of searching is used repeatedly in the news clipping service, active navigation, and Virtual Community Service applications, described below. Another use is to add a new target object quickly to the cluster tree. An existing cluster that is similar to the new target object can be located rapidly, and the new target object can be added to this cluster. If the object is beyond a certain threshold distance from the cluster center, then it is advisable to start a new cluster. Several variants of this incremental clustering scheme can be used, and can be built using variants of subroutines available in advanced statistical packages. Note that various methods can be used to locate t he new target objects that must be added to the cluster tree, depending on the architecture used. In one method, a "webcrawler" program running on a central computer periodically scans all servers in search of new target objects, calculates the target profiles of these objects, and adds them to the hierarchical cluster tree by the above method. In another, whenever a new target object is added to any of the servers, a software "agent" at that server calculates the target profile and adds it to the hierarchical cluster tree by the above method.

Rapid Profiling

In some domains, complete profiles of target objects are not always easy to construct automatically. When target objects are multimedia, for example, an attribute such as "genre" (a single textual term such as "Action", "Suspense/Thriller", "Word Games"/etc.) may be a matter of judgment and opinion, difficult to determine except by consulting a human. More significantly, if each title has an associated attribute that records the positive or negative relevance feedback to that title from various human users (consumers) then all the association scores of any newly introduced title are initially zero so that it is initially unclear what other titles are similar to the new title with respect to the users who like them. Indeed, if this associative attribute is highly weighted, the initial lack of relevance feedback information may be difficult to remedy, due to a vicious circle in which users of moderate-to-high interest are needed to provide relevance feedback but relevance feedback is needed to identify users of moderate-to-high interest.

Fortunately, however, it is often possible in principle to determine certain attributes of a new target object by extraordinary methods, including but not limited to methods that consult a human. For example, the system can in principle determine the genre of a title by consulting one more randomly chosen individual from a set of human experts, while determining the score between a new title and a particular user it can in principle show the title to that user and determine relevance feedback. Since such requests inconvenience people, however, it is important not to determine all difficult attributes this way, but only the ones that are most important is classifyng the article. "Rapid profiling" is a method for selecting those numeric attributes that are most important to determine. (Recall that all attributes can be decomposed into numeric attributes, such as association scores or term scores.) First, a set of existing target objects that already have complete or largely complete profiles are clustered using a k-means algorithm. Next, each of the resulting clusters is assigned a unique identifying number, and each clustered target object is labeled with the identifying number of its cluster. Standard methods then allow construction of a single decision tree that can determine any target object's cluster number, with substantial accuracy, by considering the attributes of the target object, one at a time. Only attributes that can if necessary be determined for any new target object are used in the construction of this decision tree. To profile a new target object, the decision tree is traversed downward from its root as far as is desired. The root of the decision tree considers some attribute of the target object. If the value of this attribute is not yet known, it is determined by a method appropriate to that attribute; for example, if the attribute is the association score of the target object with user #4589, then relevance feedback (to be used as the value of this attribute) is solicited from user #4589, perhaps by the ruse of adding the possibly uninteresting target object to a set of objects that the system recommends to the user's attention, in order to find out what the user thinks of it. Once the root attribute is determined, the rapid profiling method descends the decision tree by one level, choosing one of the decision subtrees of the root in accordance with the determined value of the root attribute. The root of this chosen subtree considers another attribute of the target object, whose value is likewise determined by an appropriate method. The process c an be repeated to determine as many attributes as desired, by whatever methods are available, although it is ordinarily stopped after a small number of attributes, to avoid the burden of determining too many attributes.

It should be noted that the rapid profiling method can be used to identify important attributes in any sort of profile, and not just profiles of target objects. In particular, recall that the disclosed method for determining topical interest through similarity requires users as well as target objects to have profiles. New users, like new target objects, may be profiled or partially profiled through the rapid profiling process. For example, when user profiles include an associative attribute that records the user's relevance feedback on all target objects in the system, the rapid profiling procedure can rapidly form a rough characterization of a new user's interests by soliciting the user's feedback on a small number of significant target objects, and perhaps also by determining a small number of other key attributes of the new user, by on-line queries, telephone surveys, or other means. Once the new user has been partially profiled in this way, the methods disclosed above predict that the new user's interests resemble the known interests of other users with similar profiles. In a variation, each user's user profile is subdivided into a set of long-term attributes, such as demographic characteristics, and a set of short-term attributes that help to identify the user's temporary desires and emotional state, such as the user's textual or multiple-choice answers to questions whose answers reflect the user's mood. A subset of the user's long-term attributes are determined when the user first registers with the system, through the use of a rapid profiling tree of long-term attributes. In addition, each time the user logs on to the system, a subset of the user's

short-term attributes are additionally determined, through the use of a separate rapid profiling tree that asks about short-term attributes.

Market Research

A technique similar to rapid profiling is of interest in market research (or voter research). Suppose that the target objects are consumers. A particular attribute in each target profile indicates whether the consumer described by that target profile has purchased product X. A decision tree can be built that attempts to determine what value a consumer has for this attribute, by consideration of the other attributes in the consumer's profile. This decision tree may be traversed to determine whether additional users are likely to purchase product X. More generally, the top few levels of the decision tree provide information, valuable to advertisers who are planning mass-market or direct-mail campaigns, about the most significant characteristics of consumers of product X.

Similar information can alternatively be extracted from a collection of consumer profiles without recourse to a decision tree, by considering attributes one at a time, and identifying those attributes on which product X's consumers differ significantly from its non-consumers. These techniques serve to characterize consumers of a particular product; they can be equally well applied to voter research or other survey research, where the objective is to characterize those individuals from a given set of surveyed individuals who favor a particular candidate, hold a particular opinion, belong to a articular demographic group, or have some other set of distinguishing attributes. Researchers may wish to purchase batches of analyzed or unanalyzed user profiles from which personal identifying information has been removed. As with any statistical database, statistical conclusions can be drawn, and relationships between attributes can be elucidated using knowledge discovery techniques which are well known in the art.

CONSUMER-BASED BETTER BUSINESS BUREAU

In the case of profiling new products, a decision tree may be useful for determining its profile quickly (for example if certain general attributes are known about the product). Rapid profiling may also be used to automatically present a selection of attributes (of at least two) with which a user selects which attribute most aptly describes the product and/or provides a weighted value of its relevance thereto. Alternatively, the decision tree presents (for each node) at least one exemplar item which the user rates indicating the degree of similarity between the system presented item(s) and the new item of interest. Additionally, for the sake of optimizing the confidence of the users being surveyed, the decision tree may also identify the user whose profiles suggest the greatest degree of similarity with the attributes or items being presented as queries. In one variation in this regard, the system selects users which are most familiar with two or more competitive products. The system performs a rapid profiling of these users, however, for product attributes which are most relevant to both products (which is produced from the result of combining or averaging both product profiles). Example attributes which are most telling about the user's perception of comparative value and quality when making a selection may include: performance, aesthetics, comfort, convenience of use, value; overall satisfaction, personal preference, as well as other relevant specific product attributes which may be determined as a part of the user's profile. By applying this technique over multiple product brands within a given category, a relative, comparative measure can be determined through averaging of results across all participating users on an attribute specific basis. Using the techniques described above which allow for pseudonymous credentialing of users or organizations by other entities, these evaluation based attributes may be automatically ascribed to each product in the form of credentials, also manually ascribed comments or descriptions may be (provided and subsequently rated by other users) to further leverage consumer participation in adding characterization attributes to a given product's or entities profile. These averaged consumer rating based credentials also act as a means of normalizing biased opinions or rogue attempts to defame a product or entity and thus are used to substantiate claims which consumers have provided and other consumers have substantiated either in the form of on-line or off-line advertisements and coupons. Comparative ratings of competitive products are achievable by targeting users which have experience with (two or more) products being compared. The most relevant attributes which both products share are presented using these rapid profiling techniques. In order to develop a truly robust statistically confident comparison across all products on an attribute by attribute basis, it is important to use this comparative product rating approach, to identify automatically which product comparisons are most statistically relevant in order to provide statistical confidence for all products being evaluated (in this comparative product context) to validation of the values of each attribute using different combinations of product comparisons is important in order to assure statistical confidence (between different users). These rated attribute credentials may also be segmented by user types using knowledge discovery techniques. For example, it is possible that users of a certain demographic, product affinity or other attribute type may have different preferences demands or expectations, thus may evaluate a product's overall quality or value (or other product attribute) differently. Additionally, these credentials may be provided as resolution credentials, for example in combination with a credential provided by a neutral third party which proves that the user is in good standing with its customers (that a "significant" number of complaints were not submitted). Brokerage exchanges which match buyers and sellers and/or act as a directory thereof may wish to apply these techniques in order to provide users with some unbiased feedback from peers about products and services being solicited peer to peer rating based resolution credentials. It is also possible to automatically present a set of survey questions to a group of users who have been previously interacting on-line with another user. Because of the subjective nature involved in characterizing individuals based upon their personal, or even professional proficiencies and weaknesses, human involvement in providing manual characterizations of a sample of users is necessary. The nature of the interaction (an associate, professional, personal, or social) may be determined through automatic means (based on the content profiles of dialogues and lists of "similar" users which they interact with) in order to automatically ascribe an associative attribute which identifies both other individuals, his/her relationship with the user and the nature of their interaction. Individuals may be automatically presented with targeted questions appropriate to the nature thereof in accordance with their mutual relationship through anticipation of which attributes or queries other individuals (like friends, associates, business partners or employers) are most likely to request in the future. These questions are ideally requested from multiple users, their values are then averaged and may be ascribed to that user as resolution credentials. In

case of disputes mediation by a judicating third party may be required. Additionally, the system may further anticipate the types of questions which are most likely to be requested by other users in the future. This approach may also be used by the system to profile skills sets, qualifications, issues of personality, character or qualification to perform a particular task. It may also direct queries to the users most likely to be qualified knowledgeable in certain popular domains, which are most likely to be relevant (and thus anticipate the types of queries that other users are likely to request. Similarly, users may be used to answer questions or provide descriptive characterizations of certain tasks or queries using rapid profiling in this way as well. Thus, tasks, (consulting on the internet, intranet, etc.) may be profiled according to the types of users who ascribe, subjective, or objective attributes to best describe the task, or attributes may be ascribed which characterize the most appropriate individuals according to their professional qualifications or other relevant attributes, such as the tasks which they have successfully performed. Accordingly, task attributes may also be conveyed to the best candidates to whom these tasks are directed. As suggested, task performance may be manually evaluated in order to provide the system with a source of performance based relevance feedback. The users who submitted the task offers are given the opportunity to provide an evaluation of the level of the quality of the work (or query response) as well as overall satisfaction regarding the response to the request offer. The requester may provide an evaluation in the form of a set of feedback comments. Additionally, the rapid profiling technique will automatically generate a set of the most relevant attributes in the form of a survey which allow the user to rate the attributes according to each relevant attribute parameter as perceived by the user. (These attributes may, of course, include those which are humanly ascribed as well). Unlike the method for automatic query routing the current system for finding optimal user skill profiles to match the particular submitted task description, the current system potentially embodies a much more complex knowledge construction requiring precision-oriented statistical knowledge about the nature of the user's numerous skill sets and the submitted tasks.

It may be very useful to use associative attributes to identify the relevant words in the task description and users who successfully provided solutions and responses to similarly described tasks in the past. According to the previously described techniques of the patent, the collection of target objects in this particular information domain include task descriptions; solutions to the requests, individuals who have provided solutions to those tasks, individuals whose profiles qualify them for solving particular problem types, and individuals who are most likely to have a need for solution to a particular type of problem. As suggested each of these types of target objects may constitute the information space of the presently described system for customized electronic identification of desirable objects. Thus in order to augment the search retrieval process the user may also be directed to potentially useful information through, menu browsing and search query navigation (and nearest neighbor, target object to target object) navigation down or across the menu as well as the current matching of appropriate users with requests are herein described. Accordingly, as relevant in the other informational domains (if the target object profiles) and the similarity between target objects is not statistically confident the system will cross correlate the statistical data from other informational domains in order to assign the most appropriate profile for each of target object for which a sparse data problem currently exists. In a more advanced embodiment, profiling of target objects in this complex domain may be further enhanced by establishing exception in the form of special appropriateness function rules between the textual, descriptive, and numeric attributes of those targeted objects (e.g. the qualification of the users, the textual attributes in the description of each task, and the evaluative description of the recipients of the task solutions provided. As in other informational domains, the exception rules which apply to a particular domain are given priority over those which apply to another domain. (Again, where cross correlation statistics are given second priority in order to maximize statistical confidence). Such exception rules may include (but are not limited to) giving special relevance between a word attribute based upon the sequence in which those textual attributes appear in the description, (or in the presence or absence of a numeric attribute in combination with a numeric attribute or a textual attribute). (These associations may also be based on their relative frequencies in the text as well) or more complex rules may be established automatically. Furthermore, if the combination of words appear, and the request is from a particular user it is likely that a particular detailed target profile is appropriate for the target object. By definition, exception rules apply exceptions in the weighting values of attributes or an attribute with an exception is present (or at least one of) at least three attributes which are present in a particular (user or target object) profile whose attribute weighting influence upon another attribute would not otherwise be recognized in a pure (non-rule based) statistical model (customized) profiles of requests which is specific to each user may be used as each user may submit similar requests in a different descriptive manner (with varying word usage). The user's needs may also vary based upon the context of what actions the user has recently performed e.g., searching through particular topics of the World Wide Web, searching through e-mail, conversing with particular users about a particular topic of engaging in these activities at certain times or in conjunction with any of the above which may indicate the context of the user's mode of activities such as work, leisure or academics. If a particular combination of words appears and it is from a particular request as part of the description of a request from a particular individual, the relevance of each attribute component of the request may be different to some degree than the request from a different individual (wherein this case these exception rules are relevant to particular users). Accordingly, the sequence of words which appear (for a particular word combination) may be suggestive of the relative importance of particular words to one another or to a particular solution or a particular individual. Accordingly in the application to matching queries or tasks with users according to their qualifications for the particular combination of qualifying credentials which a user possesses may indicate an exception rule either between particular credentials, between credentials and individual tasks (or between credentials and textual attributes in the text of task descriptions). Exception rules are not applicable for associative attributes which associate target objects users (or both) via the present similarity based techniques.

SUPPORTING ARCHITECTURE

The following section describes the preferred computer and network architecture for implementing the methods described in this patent.

Electronic Media System Architecture

FIG. 1 illustrates in block diagram form the overall architecture of an electronic media system, known in the art, in which the system for customized electronic identification

of desirable objects of the present invention can be used to provide user customized access to target objects that are available via the electronic media system. In particular, the electronic media system comprises a data communication facility that interconnects a plurality of users with a number of information servers. The users are typically individuals, whose personal computers (terminals) T_1–T_n are connected via a data communications link, such as a modem and a telephone connection established in well-known fashion, to a telecommunication network N. User information access software is resident on the user's personal computer and serves to communicate over the data communications link and the telecommunication network N with one of the plurality of network vendors V_{1-Vk} (America Online, Prodigy, CompuServe, other private companies or even universities) who provide data interconnection service with selected ones of the information servers I_1–I_m. The user can, by use of the user information access software, interact with the information servers I_1–I_m to request and obtain access to data that resides on mass storage systems –SS_m that are part of the information server apparatus. New data is input to this system y users via their personal computers T_1–T_n and by commercial information services by populating their mass storage systems SS_1–SS_m with commercial data. Each user terminal T_1–T_n and the information servers I_1–I_m have phone numbers or IP addresses on the network N which enable a data communication link to be established between a particular user terminal T_1–T_n and the selected information server I_1–I_m. A user's electronic mail address also uniquely identifies the user and the user's network vendor V_1–V_k in an industry-standard format such as: username@aol.com or username@netcom.com. The network vendors V_1–V_k provide access passwords for their subscribers (selected users), through which the users can access the information servers I_1–I_m. The subscribers pay the network vendors V_1–V_k for the access services on a fee schedule that typically includes a monthly subscription fee and usage based charges. A difficulty with this system is that there are numerous information servers I_1–I_m located around the world, each of which provides access to a set of information of differing format, content and topics and via a cataloging system that is typically unique to the particular information server I_1–I_m. The information is comprised of individual "files," which can contain audio data, video data, graphics data, text data, structured database data and combinations thereof. In the terminology of this patent, each target object is associated with a unique file: for target objects that are informational in nature and can be digitally represented, the file directly stores the informational content of the target object, while for target objects that are not stored electronically, such as purchasable goods, the file contains an identifying description of the target object. Target objects stored electronically as text files can include commercially provided news articles, published documents, letters, user-generated documents, descriptions of physical objects, or combinations of these classes of data. The organization of the files containing the information and the native format of the data contained in files of the same conceptual type may vary by information server I_1–I_m.

Thus, a user can have difficulty in locating files that contain the desired information, because the information may be contained in files whose information server cataloging may not enable the user to locate them. Furthermore, there is no standard catalog that defines the presence and services provided by all information servers I_1–I_m. A user therefore does not have simple access to information but must expend a significant amount of time and energy to excerpt a segment of the information that may be relevant to the user from the plethora of information that is generated and populated on this system. Even if the user commits the necessary resources to this task, existing information retrieval processes lack the accuracy and efficiency to ensure that the user obtains the desired information. It is obvious that within the constructs of this electronic media system, the three modules of the system for customized electronic identification of desirable objects can be implemented in a distributed manner, even with various modules being implemented on and/or by different vendors within the electronic media system. For example, the information servers I_1–I_m can include the target profile generation module while the network vendors V_1–V_k may implement the user profile generation module, the target profile interest summary generation module, and/or the profile processing module. A module can itself be implemented in a distributed manner, with numerous nodes being present in the network N, each node serving a population of users in a particular geographic area. The totality of these nodes comprises the functionality of the particular module. Various other partitions of the modules and their functions are possible and the examples provided herein represent illustrative examples and are not intended to limit the scope of the claimed invention. For the purposes of pseudonymous creation and update of users' target profile interest summaries (as described below), the vendors V_1–V_k may be augmented with some number of proxy servers, which provide a mechanism for ongoing pseudonymous access and profile building through the method described herein. At least one trusted validation server must be in place to administer the creation of pseudonyms in the system.

An important characteristic of this system for customized electronic identification of desirable objects is its responsiveness, since the intended use of the system is in an interactive mode. The system utility grows with the number of the users and this ncreases the number of possible consumer/product relationships between users and target objects. A system that serves a large group of users must maintain interactive performance and the disclosed method for profiling and clustering target objects and users can in turn be used for optimizing the distribution of data among the members of a virtual community and through a data communications network, based on users' target profile interest summaries.

Network Elements and System Characteristics

The various processors interconnected by the data communication network N as shown in FIG. 1 can be divided into two classes and grouped as illustrated in FIG. 2: clients and servers. The clients C1–Cn are individual user's computer systems which are connected to servers S1–S5 at various times via data communications links. Each of the clients Ci is typically associated with a single server Sj, but these associations can change over time. The clients C1–Cn both interface with users and produce and retrieve files to and from servers. The clients C1–Cn are not necessarily continuously on-line, since they typically serve a single user and can be movable systems, such as laptop computers, which can be connected to the data communications network N at any of a number of locations. Clients could also be a variety of other computers, such as computers and kiosks providing access to customized information as well as targeted advertising to many users, where the users identify themselves with passwords or with smart cards. A server Si is a computer system that is presumed to be continuously on-line and functions to both collect files from various sources on the data communication network N for access by

local clients C1–Cn and collect files from local clients C1–Cn for access by remote clients. The server Si is equipped with persistent storage, such as a magnetic disk data storage medium, and are interconnected with other servers via data communications links. The data communications links can be of arbitrary topology and architecture, and are described herein for the purpose of simplicity as point-to-point links or, more precisely, as virtual point-to-point links. The servers S1–S5 comprise the network vendors V1–Vk as well as the information servers I_1–I_m of FIG. 1 and the functions performed by these two classes of modules can be merged to a greater or lesser extent in a single server Si or distributed over a number of servers in the data communication network N. Prior to proceeding with the description of the preferred embodiment of the invention, a number of terms are defined. FIG. 3 illustrates in block diagram form a representation of an arbitrarily selected network topology for a plurality of servers A–D, each of which is interconnected to at least one other server and typically also to a plurality of clients p–s. Servers A–D are interconnected by a collection of point to point data communications links, and server A is connected to client r, server B is connected to clients p–q, while server D is connected to client s. Servers transmit encrypted or unencrypted messages amongst themselves: a message typically contains the textual and/or graphic information stored in a particular file, and also contains data which describe the type and origin of this file, the name of the server that is supposed to receive the message, and the purpose for which the file contents are being transmitted. Some messages are not associated with any file, but are sent by one server to other servers for control reasons, for example to request transmission of a file or to announce the availability of a new file. Messages can be forwarded by a server to another server, as in the case where server A transmits a message to server D via a relay node of either server C or servers B, C. It is generally preferable to have multiple paths through the network, with each path being characterized by its performance capability and cost to enable the network N to optimize traffic routing. In one particular implementation which is increasingly used on the World Wide Web, "channels" of content are used to enable users to select topically relevant areas of interest (e.g., National Geographic, Forbes, The Wall Street Journal, USA Today, The Disney Channel, Wired, CNN). These channels may be either accessed on demand, downloaded in advance to the user (as part of a "virtual" subscription) or selectively retrieved wherein the user's profile dictates the items selected. In this approach the items may be actively prefetched or filtered from a live chat stream. Similarly the current methods for the custom news filter may be used in this application to selectively filter and present the most relevant programming selections to the user, thus creating a "virtual channel". The basis for this concept (using a one way down stream delivery architecture) was detailed in paten pending.

In accordance with the techniques presently suggested, just as categories of information contain profiles, the most appropriate information (e.g., news information) can be automatically routed to the most appropriate category. Similarly content may be automatically routed to the most appropriate virtual channels which appeal to a particular type of audience (not only based on its content, but more subjective criteria as well) offering a unique multi media experience, writing or commentary style of its authors, etc. For this reason it may be most appropriate to initially gather relevance feedback of which users access the information in order to develop statistical confidence as to its associative attributes before it is routed to a particular channel. For example, in this regard as with the presently described techniques for customizing content through indexing, navigation and delivery from the entire scope of available information on the Internet, the scope of information may be narrowed to that of a particular channel. Additionally, because considerable overlap of content may occur between channels, authors and editors of a particular channel may use this technique to select the most desirable content from which appropriate editing and revisions may be performed as desired. These channels ideally are presented in combination with virtual communities (e.g., virtual text and voice chat rooms). They may accordingly be navigated to/from as part of the 3-D representation of the surrounding information space. For example virtual chat room associated with a news channel may incorporate scheduled live interviews with news reporters (or news makers) who had covered (or had been involved in) a particular story or combination of stories during which time participants may submit questions or comments (pseudonymously if desired). Polls may be taken about these users views on each particular event or controversial issues that are newsworthy. As suggested, preference based attributes, demographics and psychological user attributes may be statistically correlated with certain news from survey question responses or as otherwise submitted (such as in the form of active comments about that particular issue). Because questions and comments from many users may bombard a particular chat room, automated methods may be used to more efficiently manage large quantities of data. Specifically, the system may apply the following techniques:

1. Real time automatic identification of similar queries or comments which had been previously submitted (using statistical NLP or deeper NLU techniques). Once a user has submitted a question or comment, the system instantaneously indexes any similar item(s) previously submitted, automatically notifies the user that the user's submission has been canceled and automatically retrieves the previously submitted response to that previously submitted item. In the context of an ascribed posting to news groups currently known techniques such as auto-FAQ are able to generate FAQs automatically. For either live chat or (asynchronous) newsgroups, this technique may instead be used to eliminate redundancy by identifying (by indexing in real time via statistical NLP) pre-existing similar correspondences to those which are about to be initiated.
2. Automatically determine the predicted value of a user's comments and responses. This may be determined as the product of number and length of comments submitted in response to that user's postings, as well as the estimated predicted value of the response based upon the estimated value of that associated particular respondent's knowledge within the knowledge domain of the content profile of that response as well as the time that users spend reading the posting from the user's interest profile. Again, the relevance of this factor is also the product of the reader's knowledge within the knowledge domain of the content profile of the user's message. In the application to a future guest or moderator of a bulletin board or chat room (or a variation thereof called a "virtual talk show" in which the moderator fields questions by participants) the most predictively "valuable" questions, comments and/or responses are selectively prioritized for submission and reading (if a response) by the other participants. For the newsgroup application, items which are highest priority are pre-

sented first, responses to the same which are of highest priority are posted. Additionally, an item which is very similar though not as "valued" may also receive a lower value score which is less valuable though more unlike other items. In the application to live chat because the associative attribute of the list of readers of the item is unavailable (in real time) instead, the real time profiling of the message is performed and any predictive value estimated based upon that user's determined skill (value) within the knowledge domain of his/her message. Additionally, value estimation may be converted to actual price values (using the exchange of soft currency) as a variation of the price point determination scheme. In this regard, dialogues, users submitted queries, and anticipated responses thereto are appropriately matched, priced (value appraised), a "net balance" is automatically determined for each informational exchange (or transaction) and each user's "account" is debited or credited accordingly. If desired, participants external to a particular transaction may passively observe the net cost of each transaction, the price and, if the user perceives the estimated value to be inappropriate, he/she may submit a suggested modification of its value. These recommendations may be averaged in order to determine the most appropriate net transaction value. Again the relevance may be adjusted to the recommendation in accordance with the skill of that user within that knowledge domain for determining the actual modified value. This approach may be applied also on the context of Intranet (or multi-organizational Intranet).

Several applications to bandwidth content delivery may be included, including video on demand wherein video and audio programming content may be delivered to the user. Techniques for customizing program guide selections to users have been detailed in the patent pending patent entitled "System and Method for Scheduling Broadcast and Access to Video Program and Other Data" Using Customer Profiles". The present system may readily be applicable to radio programming sent over cable (or the Internet). Particularly for short programming selections like music, music video and short audio or multimedia segments, it is desirable to automate the selection process by creating a "virtual channel" of selections which are retrieved sequentially. As previously described, existing channels may be accessible to users on the WWW. These techniques for automated sequential of retrieval of content may be another implementation of another channel (e.g., using cable as a high bandwidth transmission medium to access a video server on the WWW). Another application of this architecture could be use of a client processor in a video store which receives purchases from the user's account, is maintained on the local server and the similarity measurements are processed locally or performed by a video server which may deliver high bandwidth video, audio (e.g., music) or multi media software to a compact disc at the store which is customized to the user's preferences. If user purchasing records don't yet exist or are not complete, the rapid profiling system may construct the user's profile. This system may be implemented as a stand alone credit card or smart card enabled kiosk which may be equipped with (for example) the currently described menu navigation and query techniques.

Proxy Servers and Pseudonymous Transactions

while the method of using target profile interest summaries presents many advantages to both target object providers and users, there are important privacy issues for both users and providers that must be resolved if the system is to be used freely and without inhibition by users without fear of invasion of privacy. It is likely that user s desire that some, if not all, of the user-specific information in their user profiles and target profile interest summaries remain confidential, to be disclosed only under certain circumstances related to certain types of transactions and according to their personal wishes for differing levels of confidentiality regarding their purchases and expressed interests.

However, complete privacy and inaccessibility of user transactions and profile summary information would hinder implementation of the system for customized electronic identification of desirable objects and would deprive the user of many of the advantages derived through the system's use of user-specific information. In many cases, complete and total privacy is not desired by all parties to a transaction. For example, a buyer may desire to be targeted for certain mailings that describe products that are related to his or her interests, and a seller may desire to target users who are predicted to be interested in the goods and services that the seller provides. Indeed, the usefulness of the technology described herein is contingent upon the ability of the system to collect and compare data about many users and many target objects. A compromise between total user anonymity and total public disclosure of the user's search profiles or target profile interest summary is a pseudonym. A pseudonym is an artifact that allows a service provider to communicate with users and build and accumulate records of their preferences over time, while at the same time remaining ignorant of the users' true identities, so that users can keep their purchases or preferences private. A second and equally important requirement of a pseudonym system is that it provide for digital credentials, which are used to guarantee that the user represented by a particular pseudonym has certain properties. These credentials may be granted on the basis of result of activities and transactions conducted by means of the system for customized electronic identification of desirable objects, or on the basis of other activities and transactions conducted on the network N of the present system, on the basis of users' activities outside of network N. For example, a service provider may require proof that the purchaser has sufficient funds on deposit at his/her bank, which might possibly not be on a network, before agreeing to transact business with that user. The user, therefore, must provide the service provider with proof of funds (a credential) from the bank, while still not disclosing the user's true identity to the service provider.

Our method solves the above problems by combining the pseudonym granting and credential transfer methods taught by D. Chaum and J. H. Evertse, in the paper titled "A secure and privacy-protecting protocol for transmitting personal information between organizations," with the implementation of a set of one or more proxy servers distributed throughout the network N. Each proxy server, for example S2 in FIG. 2, is a server which communicates with clients and other servers S5 in the network either directly or through anonymizing mix paths as detailed in the paper by D. Chaum titled "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms," published in Communications of the ACM, Volume 24, Number 2, February 1981. Any server in the network N may be configured to act as a proxy server in addition to its other functions. Each proxy server provides service to a set of users, which set is termed the "user base" of that proxy server. A given proxy server provides three sorts of service to each user U in its user base, as follows:

1. The first finction of the proxy server is to bidirectionally transfer communications between user U and other entities such as information servers (possibly including

the proxy server itself) and/or other users. Specifically, letting S denote the server that is directly associated with user U's client processor, the proxy server communicates with server S (and thence with user U), either through anonymizing mix paths that obscure the identity of server S and user U, in which case the proxy server knows user U only through a secure pseudonym, or else through a conventional virtual point-to-point connection, in which case the proxy server knows user U by user U's address at server S, which address may be regarded as a non-secure pseudonym for user U.

2. A second function of the proxy server is to record user-specific information associated with user U. This user-specific information includes a user profile and target profile interest summary for user U, as well as a list of access control instructions specified by user U, as described below, and a set of one-time return addresses provided by user U that can be used to send messages to user U without knowing user U's true identity. All of this user-specific information is stored in a database that is keyed by user U's pseudonym (whether secure or non-secure) on the proxy server.
3. A third function of the proxy server is to act as a selective forwarding agent for unsolicited communications that are addressed to user U: the proxy server forwards some such communications to user U and rejects others, in accordance with the access control instructions specified by user U.

Our combined method allows a given user to use either a single pseudonym in all transactions where he or she wishes to remain pseudonymous, or else different pseudonyms for different types of transactions. In the latter case, each service provider might transact with the user under a different pseudonym for the user. More generally, a coalition of service providers, all of whom match users with the same genre of target objects, might agree to transact with the user using a common pseudonym, so that the target profile interest summary associated with that pseudonym would be complete with respect to said genre of target objects. When a user employs several pseudonyms in order to transact with different coalitions of service providers, the user may freely choose a proxy server to service each pseudonym; these proxy servers may be the same or different.

From the service provider's perspective, our system provides security, in that it can guarantee that users of a service are legitimately entitled to the services used and that no user is using multiple pseudonyms to communicate with the same provider. This uniqueness of pseudonyms is important for the purposes of this application, since the transaction information gathered for a given individual must represent a complete and consistent picture of a single user's activities with respect to a given service provider or coalition of service providers; otherwise, a user's target profile interest summary and user profile would not be able to represent the user's interests to other parties as completely and accurately as possible.

The service provider must have a means of protection from users who violate previously agreed upon terms of service. For example, if a user that uses a given pseudonym engages in activities that violate the terms of service, then the service provider should be able to take action against the user, such as denying the user service and blacklisting the user from transactions with other parties that the user might be tempted to defraud. This type of situation might occur when a user employs a service provider for illegal activities or defaults in payments to the service provider. The method of the paper titled "Security without identification: Transaction systems to make Big-Brother obsolete", published in the Communications of the ACM, 28(10), October 1985; pp.1030–1044, incorporated herein, provides for a mechanism to enforce protection against this type of behavior through the use of resolution credentials, which are credentials that are periodically provided to individuals contingent upon their behaving consistent with the agreed upon terms of service between the user and information provider and network vendor entities (such as regular payment for services rendered, civil conduct, etc.). For the user's safety, if the issuer of a resolution credential refuses to grant this resolution credential to the user, then the refusal may be appealed to an adjudicating third party. The integrity of the user profiles and target profile interest summaries stored on proxy servers is important: if a seller relies on such user-specific information to deliver promotional offers or other material to a particular class of users, but not to other users, then the user-specific information must be accurate and untampered with in any way. The user may likewise wish to ensure that other parties not tamper with the user's user profile and target profile interest summary, since such modification could degrade the system's ability to match the user with the most appropriate target objects. This is done by providing for the user to apply digital signatures to the control messages sent by the user to the proxy server. Each pseudonym is paired with a public cryptographic key and a private cryptographic key, where the private key is known only to the user who holds that pseudonym; when the user sends a control message to a proxy server under a given pseudonym, the proxy server uses the pseudonym's public key to verify that the message has been digitally signed by someone who knows the pseudonym's private key. This prevents other parties from masquerading as the user.

Our approach, as disclosed in this application, provides an improvement over the prior art in privacy-protected pseudonymny for network subscribers such as taught in U.S. Pat. No. 5,245,656, which provides for a name translator station to act as an intermediary between a service provider and the user. However, while U.S. Pat. No. 5,245,656 provides that the information transmitted between the end user U and the service provider be doubly encrypted, the fact that a relationship exists between user U and the service provider is known to the name translator, and this fact could be used to compromise user U, for example if the service provider specializes in the provision of content that is not deemed acceptable by user U's peers. The method of U.S. Pat. No. 5,245,656 also omits a method for the convenient updating of pseudonymous user profile information, such as is provided in this application, and does not provide for assurance of unique and credentialed registration of pseudonyms from a credentialing agent as is also provided in this application, and does not provide a means of access control to the user based on profile information and conditional access as will be subsequently described. The method described by Loeb et al. also does not describe any provision for credentials, such as might be used for authenticating a user's right to access particular target objects, such as target objects that are intended to be available only upon payment of a subscription fee, or target objects that are intended to be unavailable to younger users.

Proxy Server Description

In order that a user may ensure that some or all of the information in the user's user profile and target profile interest summary remain dissociated from the user's true identity, the user employs as an intermediary any one of a number of proxy servers available on the data communication network N of FIG. 2 (for example, server S2). The

proxy servers function to disguise the true identity of the user from other parties on the data communication network N. The proxy server represents a given user to either single network vendors and information servers or coalitions thereof A proxy server, e.g. S2, is a server computer with CPU, main memory, secondary disk storage and network communication function and with a database function which retrieves the target profile interest summary and access control instructions associated with a particular pseudonym P, which represents a particular user U, and performs bi-directional routing of commands, target objects and billing information between the user at a given client (e.g. C3) and other network entities such as network vendors V1–Vk and information servers I1–Im. Each proxy server maintains an encrypted target profile interest summary associated with each allocated pseudonym in its pseudonym database D. The actual user-specific information and the associated pseudonyms need not be stored locally on the proxy server, but may alternatively be stored in a distributed fashion and be remotely addressable from the proxy server via point-to-point connections.

The proxy server supports two types of bi-directional connections: point-to-point connections and pseudonymous connections through mix paths, as taught by D. Chaum in the paper titled "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, Volume 24, Number 2, February 1981. The normal connections between the proxy server and information servers, for example a connection between proxy server S2 and information server S4 in FIG. 2, are accomplished through the point-to-point connection protocols provided by network N as described in the "Electronic Media System Architecture" section of this application. The normal type of point-to-point connections may be used between S2–S4, for example, since the dissociation of the user and the pseudonym need only occur between the client C3 and the proxy server S2, where the pseudonym used by the user is available. Knowing that an information provider such as S4 communicates with a given pseudonym P on proxy server S2 does not compromise the true identity of user U. The bidirectional connection between the user and the proxy server S2 can also be a normal point-to-point connection, but it may instead be made anonymous and secure, if the user desires, though the consistent use of an anonymizing mix protocol as taught by D. Chaum in the paper titled "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, Volume 24, Number 2, February 1981. This mix procedure provides untraceable secure anonymous mail between to parties with blind return addresses through a set of forwarding and return routing servers termed "mixes". The mix routing protocol, as taught in the Chaum paper, is used with the proxy server S2 to provide a registry of persistent secure pseudonyms that can be employed by users other than user U, by information providers I1–Im, by vendors V1–Vk and by other proxy servers to communicate with the users in the proxy server's user base on a continuing basis. The security provided by this mix path protocol is distributed and resistant to traffic analysis attacks and other known forms of analysis which may be used by malicious parties to try and ascertain the true identity of a pseudonym bearer. Breaking the protocol requires a large number of parties to maliciously collude or be cryptographically compromised. In addition an extension to the method is taught where the user can include a return path definition in the message so the information server S4 can return the requested information to the user's client processor C3. We utilize this feature in a novel fashion to provide for access and reachability control under user and proxy server control.

Validation and Allocation of a Unique Pseudonym

Chaum's pseudonym and credential issuance system, as described in a publication by D. Chaum and J. H. Evertse, titled "A secure and privacy-protecting protocol for transmitting personal information between organizations," has several desirable properties for use as a component in our system. The system allows for individuals to use different pseudonyms with different organizations (such as banks and coalitions of service providers). The organizations which are presented with a pseudonym have no more information about the individual than the pseudonym itself and a record of previous transactions carried out under that pseudonym. Additionally, credentials, which represent facts about a pseudonym that an organization is willing to certify, can be granted to a particular pseudonym, and transferred to other pseudonyms that the same user employs. For, example, the user can use different pseudonyms with different organizations (or disjoint sets of organizations), yet still present credentials that were granted by one organization, under one pseudonym, in order to transact with another organization under another pseudonym, without revealing that the two pseudonyms correspond to the same user. Credentials may be granted to provide assurances regarding the pseudonym bearer's age, financial status, legal status, and the like. For example, credentials signifying "legal adult" may be issued to a pseudonym based on information known about the corresponding user by the given is suing organization. Then, when the credential is transferred to another pseudonym that represents the user to another disjoint organization, presentation of this credential on the other pseudonym can be taken as proof of legal adulthood, which might satisfy a condition of terms of service. Credential-issuing organizations may also certify particular facts about a user's demographic profile or target profile interest summary, for example by granting a credential that asserts "the bearer of this pseudonym is either well-read or is middle-aged and works for a large company"; by presenting this credential to another entity, the user can prove eligibility for (say) a discount without revealing the user 's personal data to that entity.

Additionally, the method taught by Chaum provides for assurances that no individual may correspond with a given organization or coalition of organizations using more than one pseudonym; that credentials may not be feasibly forged by the user; and t hat credentials may not be transferred from one user's pseudonym to a different user's pseudonym. Finally, the method provides for expiration of credentials and for the issuance of "black marks" against Individuals who do not act according to the terms of service that they are extended. This is done through the resolution credential mechanism as described in Chaum's work, in which resolutions are issued periodically by organizations to pseudonyms that are in good standing. If a user is not issued this resolution credential by a particular organization or coalition of organization, then this user cannot have it available to be transferred to other pseudonyms which he uses with other organizations. Therefore, the user cannot convince these other organizations that he has acted accordance with terms of service in other dealings. If this is the case, then the organization can use this lack of resolution credential to infer that the user is not in good standing in his other dealings. In one approach organizations (or other users) may issue a list of quality related credentials based upon the experience of transaction (or interaction) with the user which may act similarly to a letter of recommendation as in a resume. If such a credential is issued from multiple

organizations, their values become averaged. In an alternative variation organizations may be issued credentials from users such as customers which may be used to indicate to other future users quality of service which can be expected by subsequent users on the basis of various criteria. In one approach, the system automatically generated the primary attributes contained in the profile of the user or organization. Each attribute is then appropriately rated in order to become a list of quality related credentials.

In our implementation, a pseudonym is a data record consisting of two fields. The first field specifies the address of the proxy server at which the pseudonym is registered. The second field contains a unique string of bits (e.g., a random binary number) that is associated with a particular user; credentials take the form of public-key digital signatures computed on this number, and the number itself is issued by a pseudonym administering server Z, as depicted in FIG. 2, and detailed in a generic form in the paper by D. Chaum and J. H. Evertse, titled "A secure and privacy-protecting protocol for transmitting personal information between organizations.". It is possible to send information to the user holding a given pseudonym, by enveloping the information in a control message that specifies the pseudonym and is addressed to the proxy server that is named in the first field of the pseudonym; the proxy server may forward the information to the user upon receipt of the control message.

While the user may use a single pseudonym for all transactions, in the more general case a user has a set of several pseudonyms, each of which represents the user in his or her interactions with a single provider or coalition of service providers. Each pseudonym in the pseudonym set is designated for transactions with a different coalition of related service providers, and the pseudonyms used with one provider or coalition of providers cannot be linked to the pseudonyms used with other disjoint coalitions of providers. All of the user's transactions with a given coalition can be linked by virtue of the fact that they are conducted under the same pseudonym, and therefore can be combined to define a unified picture, in the form of a user profile and a target profile interest summary, of the user's interests vis-a-vis the service or services provided by said coalition. There are other circumstances for which the use of a pseudonym may be useful and the present description is in no way intended to limit the scope of the claimed invention for example, the previously described rapid profiling tree could be used to pseudonymously acquire information about the user which is considered by the user to be sensitive such as that information which is of interest to such entitites as insurance companies, medical specialists, family counselors or dating services.

Detailed Protocol

In our system, the organizations that the user U interacts with are the servers S1–Sn on the network N. However, rather than directly corresponding with each server, the user employs a proxy server, e.g. S2, as an intermediary between the local server of the user's own client and the information provider or network vendor. Mix paths as described by D. Chaum in the paper titled "Untraceable Electronic Mail, Return Addresses, and Digital Pseudonyms", Communications of the ACM, Volume 24, Number 2, February 1981 allow for untraceability and security between the client, such as C3, and the proxy server, e.g. S2. Let S(M,K) represent the digital signing of message M by modular exponentiation with key K as detailed in a paper by Rivest, R. L., Shamir, A., and Adleman, L. Titled "A method for obtaining digital signatures and public-key cryptosystems", published in the Comm. ACM 21, February 2, 120–126. Once a user applies to server Z for a pseudonym P and is granted a signed pseudonym signed with the private key SK_Z of server Z, the following protocol takes place to establish an entry for the user U in the proxy server S2's database D. 1. The user now sends proxy server S2 the pseudonym, which has been signed by Z to indicate the authenticity and uniqueness of the pseudonym. The user also generates a PK_p, SK_p key pair for use with the granted pseudonym; where is the private key associated with the pseudonym and PK_p is the public key associated with the pseudonym. The user forms a request to establish pseudonym P on proxy server S2, by sending the signed pseudonym $S(P, SK_Z)$ to the proxy server S2 along with a request to create a new database entry, indexed by P, and the public key PK_p. It envelopes the message and transmits it to a proxy server S2 through an anonymizing mix path, along with an anonymous return envelope header. 2. The proxy server S2 receives the database creation entry request and associated certified pseudonym message. The proxy server S2 checks to ensure that the requested pseudonym P is signed by server Z and if so grants the request and creates a database entry for the pseudonym, as well as storing the user's public key PK_p to ensure that only the user U can make requests in the future using pseudonym P. 3. The structure of the user's database entry consists of a user profile as detailed herein, a target profile interest summary as detailed herein, and a Boolean combination of access control criteria as detailed below, along with the associated public key for the pseudonym P. 4. At any time after database entry for Pseudonym P is established, the user U may provide proxy server S2 with credentials on that pseudonym, provided by third parties, which credentials make certain assertions about that pseudonym. The proxy server may verify those credentials and make appropriate modifications to the user's profile as required by these credentials. such as recording the user's new demographic status as an adult. It may also store those credentials, so that it can present them to service providers on the user's behalf

The above steps may be repeated, with either the same or a different proxy server, each time user U requires a new pseudonym for use with a new and disjoint coalition of providers. In practice there is an extremely small probability that a given pseudonym may have already been allocated by due to the random nature of the pseudonym generation process carried out by Z. If this highly unlikely event occurs, then the proxy server S2 may reply to the user with a signed message indicating that the generated pseudonym has already been allocated, and asking for a new pseudonym to be generated.

Pseudonymous Control of an Information Server

Once a proxy server S2 has authenticated and registered a user's pseudonym, the user may begin to use the services of the proxy server S2, in interacting with other network entities such as service providers, as exemplified by server S4 in FIG. 2, an information service provider node connected to the network. The user controls the proxy server S2 by forming digitally encoded requests that the user subsequently transmits to the proxy server S2 over the network N. The nature and format of these requests will vary, since the proxy server may be used for any of the services described in this application, such as the browsing, querying, and other navigational functions described below.

In a generic scenario, the user wishes to communicate under pseudonym P with a particular information provider or user at address A, where P is a pseudonym allocated to the user and A is either a public network address at a server such as S4, or another pseudonym that is registered on a proxy

server such as S4. (In the most common version of this scenario, address A is the address of an information provider, and the user is requesting that the in formation provider send target objects of interest.) The user must form a request R to proxy server S2, that requests proxy server S2 to send a message to address A and to forward the response back to the user. The user may thereby communicate with other parties, either non-pseudonymous parties, in the case where address A is a public network address, or pseudonymous parties, in the case where address A is a pseudonym held by, for example, a business or another user who prefers to operate pseudonymously.

In other scenarios, the request R to proxy server S2 formed by the user may have different content. For example, request R may instruct proxy server S2 to use the methods described later in this description to retrieve from the most convenient server a particular piece of information that has been multicast to many servers, and to send this information to the user. Conversely, request R may instruct proxy server S2 to multicast to many servers a file associated with a new target object provided by the user, as described below. If the user is a subscriber to the news clipping service described below, request R may instruct proxy server S2 to forward to the user all target objects that the news clipping service has sent to proxy server S2 for the user's attention. If the user is employing the active navigation service described below, request R may instruct proxy server S2 to select a particular cluster from the hierarchical cluster tree and provide a menu of its subclusters to the user, or to activate a query that temporarily affects proxy server S2's record of the user's target profile interest summary. If the user is a member of a virtual community as described below, request R may instruct proxy server S2 to forward to the user all messages that have been sent to the virtual community.

Regardless of the content of request R, the user, at client C3, initiates a connection to the user's local server S1, and instructs server S1 to send the request R along a secure mix path to the proxy server S2, initiating the following sequence of actions:

1. The user's client processor C3 forms a signed message $S(R, SK_p)$, which is paired with the user's pseudonym P and (if the request R requires a response) a secure one-time set of return envelopes, to form a message M. It protects the message M with an multiply enveloped route for the outgoing path. The enveloped route s provide for secure communication between S1 and the proxy server S2. The message M is enveloped in the most deeply nested message and is therefore difficult to recover should the message be intercepted by an eavesdropper.
2. The message M is sent by client C3 to its local server S1, and is then routed by the data communication network N from server S1 through a set of mixes as dictated by the outgoing envelope set and arrives at the selected proxy server S2.
3. The proxy server S2 separates the received message M into the request message R, the pseudonym P, and (if included) the set of envelopes for the return path. The proxy server S2 uses pseudonym P to index and retrieve the corresponding record in proxy server S2's database, which record is stored in local storage at the proxy server S2 or on other distributed storage media accessible to proxy server S2 via the network N. This record contains a public key PK_p, user-specific information, and credentials associated with pseudonym P. The proxy server S2 uses the public key PK_p to check that the signed version $S(R, SK_p)$ of request message R is valid.
4. Provided that the signature on request message R is valid, the proxy server S2 acts on the request R. For example, in the generic scenario described above, request message R includes an embedded message M1 and an address A to whom message M1 should be sent; in this case, proxy server S2 sends message M1 to the server named in address A, such as server S4. The communication is done using signed and optionally encrypted messages over the normal point to point connections provided by the data communication network N. When necessary in order to act on embedded message M1, server S4 may exchange or be caused to exchange further signed and optionally encrypted messages with proxy server S2, still over normal point to point connections, in order to negotiate the release of user-specific information and credentials from proxy server S2. In particular, server S4 may require server S2 to supply credentials proving that the user is entitled to the information requested—for example, proving that the user is a subscriber in good standing to a particular information service, that the user is old enough to legally receive adult material, and that the user has been offered a particular discount (by means of a special discount credential issued to the user's pseudonym). Such a special discount credential may be automatically provided by a trusted process residing in the proxy server i.e. the price point algorithm. In one approach, this special discount credential may persist so long as the trusted process on the proxy server allows it to (that provides access to an appropriate discount by that user, this may be termed "digital coupon"). In another variation, the terms of the special discount credential may vary in accordance with certain user actions (which are pre-specified to the user) e.g. automatically modifying the degree or nature of the discount in response to user purchasing behavior towards that vendor or product (or jointly marketed products or a vendor consortium). This may be termed a "digital shopper's card".
5. If proxy server S2 has sent a message to a server S4 and server S4 has created a response M2 to message M1 to be sent to the user, then server S4 transmits the response M2 to the proxy server S2 using normal network point-o-point connections.
6. The proxy server S2, upon receipt of the response M2, creates a return message Mr comprising the response M2 embedded in the return envelope set that was earlier transmitted to proxy server S2 by the user in the original message M. It transmits the return message Mr along the pseudonymous mix path specified by this return envelope set, so that the response M2 reaches the user at the user's client processor C3.
7. The response M2 may contain a request for electronic payment to the information server S4. The user may then respond by means of a message M3 transmitted by the same means as described for message M1 above, which message M3 encloses some form of anonymous payment. Alternatively, the proxy server may respond automatically with such a payment, which is debited from an account maintained by the proxy server for this user.
8. Either the response message M2 from the information server S4 to the user, or a subsequent message sent by the proxy server S2 to the user, may contain advertising material that is related to the user's request and/or is targeted to the user. Typically, if the user has just retrieved a target object X, then (a) either proxy server

S2 or information server S4 determines a weighted set of advertisements that are "associated with" target object X (b) a subset of this set is chosen randomly, where the weight of an advertisement is proportional to the probability that it is included in the subset, and (c) proxy server S2 selects from this subset just those advertisements that the user is most likely to be interested in. In the variation where proxy server S2 determines the set of advertisements associated with target object X, then this set typically consists of all advertisements that the proxy server's owner has been paid to disseminate and whose target profiles are within a threshold similarity distance of the target profile of target object X. In the variation where proxy server S4 determines the set of advertisements associated with target object X advertisers typically purchase the right to include advertisements in this set. In either case, the weight of an advertisement is determined by the amount that an advertiser is willing to pay. Following step (c), proxy server S2 retrieves the selected advertising material and transmits it to the user's client processor C3, where it will be displayed to the user, within a specified length of time after it is received, by a trusted process running on the user's client processor C3. When proxy server S2 transmits an advertisement, it sends a message to the advertiser, indicating that the advertisement has been transmitted to a user with a particular predicted level of interest. The message may also indicate the identity of target object X. In return, the advertiser may transmit an electronic payment to proxy server S2; proxy server S2 retains a service fee for itself, optionally forwards a service fee to information server S4, and the balance is forwarded to the user or used to credit the user's account on the proxy server.

9. If the response M2 contains or identifies a target object, the passive and/or active relevance feedback that the user provides on this object is tabulated by a process on the user's client processor C3. A summary of such relevance feedback information, digitally signed by client processor C3 with a proprietary private key SK_{C3}, is periodically transmitted through an a secure mix path to the proxy server S2, whereupon the search profile generation module **202** resident on server S2 updates the appropriate target profile interest summary associated with pseudonym P, provided that the signature on the summary message can be authenticated with the corresponding public key PK_{C3} which is available to all tabulating process that are ensured to have integrity.

When a consumer enters into a financial relationship with a particular information server based on both parties agreeing to terms for the relationship, a particular pseudonym may be extended for the consumer with respect to the given provider as detailed in the previous section. When entering into such a relationship, the consumer and the service provider agree to certain terms. However, if the user violates the terms of this relationship, the service provider may decline to provide service to the pseudonym under which it transacts with the user. In addition, the service provider has the recourse of refusing to provide resolution credentials to the pseudonym, and may choose to do so until the pseudonym bearer returns to good standing.

Pre-Fetching of Target Objects

In some circumstances, a user may request access in sequence to many files, which are stored on one or more information servers. This behavior is common when navigating a hypertext system such as the World Wide Web, or when using the target object browsing system described below.

In general, the user requests access to a particular target object or menu of target objects; once the corresponding file has been transmitted to the user's client processor, the user views its contents and makes another such request, and so on. Each request may take many seconds to satisfy, due to retrieval and transmission delays. However, to the extent that the sequence of requests is predictable, the system for customized electronic identification of desirable objects can respond more quickly to each request, by retrieving or starting to retrieve the appropriate files even before the user requests them. This early retrieval is termed "pre-fetching of files." As earlier suggested the present system also enables users to view automatically ranked hyperlinks in accordance with their relative priority to the user profile. By combining this approach with prefetching (suggesting to the user for files prefetching has already been initiated) overall prediction of the next user action is further enhanced.

Pre-fetching of locally stored data has been heavily studied in memory hierarchies, including CPU caches and secondary storage (disks), for several decades. A leader in this area has been A. J. Smith of Berkeley, who identified a variety of schemes and analyzed opportunities using extensive traces in both databases and CPU caches. His conclusion was that general schemes only really paid off where there was some reasonable chance that sequential access was occurring, e.g., in a sequential read of data. As the balances between various latencies in the memory hierarchy shifted during the late 1980's and early 1990's, J. M. Smith and others identified further opportunities for pre-fetching of both locally stored data and network data. In particular, deeper analysis of patterns in work by Blaha showed the possibility of using expert systems for deep pattern analysis that could be used for pre-fetching. Work by J. M. Smith proposed the use of reference history trees to anticipate references in storage hierarchies where there was some historical data. Recent work by Touch and the Berkeley work addressed the case of data on the World-Wide Web, where the large size of images and the long latencies provide extra incentive to pre-fetch; Touch's technique is to pre-send when large bandwidths permit some speculation using HTML storage references embedded in WEB pages, and the Berkeley work uses techniques similar to J. M. Smith's reference histories specialized to the semantics of HTML data.

Successful pre-fetching depends on the ability of the system to predict the next action or actions of the user. In the context of the system for customized electronic identification of desirable objects, it is possible to cluster users into groups according to the similarity of their user profiles. Any of the well-known pre-fetching methods that collect and utilize aggregate statistics on past user behavior, in order to predict future user behavior, may then be implemented in so as to collect and utilize a separate set of statistics for each cluster of users. In this way, the system generalizes its access pattern statistics from each user to similar users, without generalizing among users who have substantially different interests. The system may further collect and utilize a similar set of statistics that describes the aggregate behavior of all users; in cases where the system cannot confidently make a prediction as to what a particular user will do, because the relevant statistics concerning that user's user cluster are derived from only a small amount of data, the system may instead make its predictions based on the aggregate statistics for all users, which are derived from a larger amount of data. For the sake of concreteness, we now describe a particular instantiation of a pre-fetching system, that both employs these insights and that makes its pre-fetching decisions

through accurate measurement of the expected cost and benefit of each potential pre-fetch.

Pre-fetching exhibits a cost-benefit tradeoff. Let t denote the approximate number of minutes that pre-fetched files are retained in local storage (before they are deleted to make room for other pre-fetched files). If the system elects to pre-fetch a file corresponding to a target object X, then the user benefits from a fast response at no extra cost, provided that the user explicitly requests target object X soon thereafter. However, if the user does not request target object X within t minutes of the pre-fetch, then the pre-fetch was worthless, and its cost is an added cost that must be borne (directly or indirectly) by the user. The first scenario therefore provides benefit at no cost, while the second scenario incurs a cost at no benefit. The system tries to favor the first scenario by pre-fetching only those files that the user will access anyway. Depending on the user's wishes, the system may pre-fetch either conservatively, where it controls costs by pre-fetching only files that the user is extremely likely to request explicitly (and that are relatively cheap to retrieve), or more aggressively, where it also pre-fetches files that the user is only moderately likely to request explicitly, thereby increasing both the total cost and (to a lesser degree) the total benefit to the user.

In the system described herein, pre-fetching for a user U is accomplished by the user's proxy server S. Whenever proxy server S retrieves a user-requested file F from an information server, it uses the identity of this file F and the characteristics of the user, as described below, to identify a group of other files G1 . . . Gk that the user is likely to access soon. The user's request for file F is said to "trigger" files G1 . . . Gk. Proxy server S pre-fetches each of these triggered files Gi as follows:

1. Unless file Gi is already stored locally (e.g., due to previous pre-fetch), proxy server S retrieves file Gi from an appropriate information server and stores it locally.
2. Proxy server S timestamps its local copy of file Gi as having just been pre-fetched, so that file Gi will be retained in local storage for a minimum of approximately t minutes before being deleted.

Whenever user U (or, in principle, any other user registered with proxy server S) requests proxy server S to retrieve a file that has been pre-fetched and not yet deleted, proxy server S can then retrieve the file from local storage rather than from another server. In a variation on steps 1–2 above, proxy server S pre-fetches a file Gi somewhat differently, so that pre-fetched files are stored on the user's client processor q rather than on server S:

1. If proxy server S has not pre-fetched file Gi in the past t minutes, it retrieves file Gi and transmits it to user U'sclient processor q.
2. Upon receipt of the message sent in step 1, client q stores a local copy of file Gi if one is not currently stored.
3. Proxy server S notifies client q that client q should timestamp its local copy of file Gi; this notification may be combined with the message transmitted in step 1, if any.
4. Upon receipt of the message sent in step 3, client q timestamps its local copy of file Gi as having just been pre-fetched, so that file Gi will be retained in local storage for a minimum of approximately t minutes before being deleted.

During the period that client q retains file Gi in local storage, client q can respond to any request for file Gi (by user U or, in principle, any other user of client q) immediately and without the assistance of proxy server S.

The difficult task is for proxy server S, each time it retrieves a file F in response to a request, to identify the files G1 . . . Gk that should be triggered by the request for file F and pre-fetched immediately. Proxy server S employs a cost-benefit analysis, performing each pre-fetch whose benefit exceeds a user-determined multiple of its cost; the user may set the multiplier low for aggressive prefetching or high for conservative prefetching. These pre-fetches may be performed in parallel. The benefit of pre-fetching file Gi immediately is defined to be the expected number of seconds saved by such a pre-fetch, as compared to a situation where Gi is left to be retrieved later (either by a later pre-fetch, or by the user's request) if at all. The cost of pre-fetching file Gi immediately is defined to be the expected cost for proxy server S to retrieve file Gi, as determined for example by the network locations of server S and file Gi and by information provider charges, times 1 minus the probability that proxy server S will have to retrieve file Gi within t minutes (to satisfy either a later pre-fetch or the user's explicit request) if it is not pre-fetched now.

The above definitions of cost and benefit have some attractive properties. For example, if users tend to retrieve either file F1 or file F2 (say) after file F, and tend only in the former case to subsequently retrieve file G1, then the system will generally not pre-fetch G1 immediately after retrieving file F: for, to the extent that the user is likely to retrieve file F2, the cost of the pre-fetch is high, and to the extent that the user is likely to retrieve file F1 instead, the benefit of the pre-fetch is low, since the system can save as much or nearly as much time by waiting until the user chooses F1 and pre-fetching G1 only then.

The proxy server S may estimate the necessary costs and benefits by adhering to the following discipline:

1. Proxy server S maintains a set of disjoint clusters of the users in its user base, clustered according to their user profiles.
2. Proxy server S maintains an initially empty set PFT of "pre-fetch triples" <C,F,G>, where F and G are files, and where C identifies either a cluster of users or the set of all users in the user base of proxy server S. Each pre-fetch triple in the set PFT is associated with several stored values specific to that triple. Pre-fetch triples and their associated values are maintained according to the rules in 3 and 4.
3. Whenever a user U in the user base of proxy server S makes a request R2 for a file G, or a request R2 that triggers file G, then proxy server S takes the following actions:
   a. For C being the user cluster containing user U, and then again for C being the set of all users:
   b. For any request R0 for a file, say file F, made by user U during the t minutes strictly prior to the request R2:
   c. If the triple <C,F,G> is not currently a member of the set PFT, it is added to the set PFT with a count of 0, a trigger-count of 0, a target-count of 0, a total benefit of 0, and a timestamp whose value is the current date and time.
   d. The count of the triple <C,F,G> is increased by one.
   e. If file G was not triggered or explicitly retrieved by any request that user U made strictly in between requests R0 and R2, then the target-count of the triple <C,F,G> is increased by one.
   f. If request R2 was a request for file G, then the total benefit of triple <C,F,G> is increased either by the

time elapsed between request R0 and request R2, or by the expected time to retrieve file G, whichever is less.

g. If request R2 was a request for file G, and G was triggered or explicitly retrieved by one or more requests that user U made strictly in between requests R0 and R2, with R1 denoting the earliest such request, then the total benefit of triple <C,F,G> is decreased either by the time elapsed between request R1 and request R2, or by the expected time to retrieve file G, whichever is less.

4. If a user U requests a file F, then the trigger-count is incremented by one for each triple currently in the set PFT such that the triple has form <C,F,G>, where user U is in the set or cluster identified by C.

5. The "age" of a triple <C,F,G> is defined to be the number of days elapsed between its timestamp and the current date and time. If the age of any triple <C,F,G> exceeds a fixed constant number of days, and also exceeds a fixed constant multiple of the triple's count, then the triple may be deleted from the set PFT.

Proxy server S can therefore decide rapidly which files G should be triggered by a request for a given file F from a given user U, as follows.

1. Let C0 be the user cluster containing user U, and C1 be the set of all users.

2. Server S constructs a list L of all triples <C0,F,G> such that <C0,F,G> appears in set PFT with a count exceeding a fixed threshold.

3. Server S adds to list L all triples <C1,F,G> such that <C0,F,G> does not appear on list L and <C1,F,G> appears in set PFT with a count exceeding another fixed threshold.

4. For each triple <C,F,G> on list L:

5. Server S computes the cost of triggering file G to be expected cost of retrieving file Gi, times 1 minus the quotient of the target-count of <C,F,G> by the trigger-count of <C,F,G>.

6. Server S computes the benefit of triggering file G to be the total benefit of <C,F,G> divided by the count of <C,F,G>.

7. Finally, proxy server S uses the computed cost and benefit, as described earlier, to decide whether file G should be triggered. The approach to pre-fetching just described has the advantage that all data storage and manipulation concerning pre-fetching decisions by proxy server S is handled locally at proxy server S. However, this "user-based" approach does lead to duplicated storage and effort across proxy servers, as well as incomplete data at each individual proxy server. That is, the information indicating what files are frequently retrieved after file F is scattered in an uncoordinated way across numerous proxy servers. An alternative, "file-based" approach is to store all such information with file F itself The difference is as follows. In the user-based approach, a pre-fetch triple <C,F,G> in server S's set PFT may mention any file F and any file G on the network, but is restricted to clusters C that are subsets of the user base of server S. By contrast, in the file-based approach, a pre-fetch triple <C,F,G> in server S's set PFT may mention any user cluster C and any file G on the network, but is restricted to files F that are stored on server S. (Note that in the file-based approach, user clustering is network wide, and user clusters may include users from different proxy servers.) When a proxy server S2 sends a request to server S to retrieve file F for a user U, server S2 indicates in this message the user U's user cluster C0, as well as the user U's value for the user-determined multiplier that is used in cost-benefit analysis. Server S can use this information, together with all its triples in its set PFT of the form <C0,F,G> and <C1,F,G>, where C1 is the set of all users everywhere on the network, to determine (exactly as in the user-based approach) which files G1 . . . Gk are triggered by the request for file F. When server S sends file F back to proxy server S2, it also sends this list of files G1. . . Gk, so that proxy server S2 can proceed to pre-fetch files G1 . . . Gk.

The file-based approach requires some additional data transmission. Recall that under the user-based approach, server S must execute steps 3c–3g above for any ordered pair of requests R0 and R2 made within t minutes of each other by a user who employs server S as a proxy server. Under the file-based approach, server S must execute steps 3c–3g above for any ordered pair of requests R0 and R2 made within t minutes of each other, by any user on the network, such that R0 requests a file stored on server S. Therefore, when a user makes a request R2, the user's proxy server must send a notification of request R2 to all servers S such that, during the preceding t minutes (where the variable t may now depend on server S), the user has made a request R0 for a file stored on server S. This notification need not be sent immediately, and it is generally more efficient for each proxy server to buffer up such notifications and send them periodically in groups to the appropriate servers.

Access And Reachability Control of Users and User-Specific Information

Although users' true identities are protected by the use of secure mix paths, pseudonymity does not guarantee complete privacy. In particular, advertisers can in principle employ user-specific data to barrage users with unwanted solicitations. The general solution to this problem is for proxy server S2 to act as a representative on behalf of each user in its user base, permitting access to the user and the user's private data only in accordance with criteria that have been set by the user. Proxy server S2 can restrict access in two ways:

1. The proxy server S2 may restrict access by third parties to server S2's pseudonymous database of user-specific information. When a third party such as an advertiser sends a message to server S2 requesting the release of user-specific information for a pseudonym P, server S2 re fuses to honor the request unless the message includes credentials for the accessor adequate to prove that the accessor is entitled to this information. The user associated with pseudonym P may at any time send signed control messages to proxy server S2, specifying the credentials or Boolean combinations of credentials that proxy server S2 should thenceforth consider to be adequate grounds for releasing a specified subset of the information associated with pseudonym P. Proxy server S2 stores these access criteria with its database record for pseudonym P. For example, a user might wish to proxy server S2 to release purchasing information only to selected information providers, to charitable organizations (that is, organizations that can provide a government-issued credential that is issued only to registered charities), and to market researchers who have paid user U for the right to study user U's purchasing habits.

2. The proxy server S2 may restrict the ability of third parties to send electronic messages to the user. When a third party such as an advertiser attempts to send information (such as a textual message or a request to enter into spoken or written real -time communication) to pseudonym P, by sending a message to proxy server S2 requesting proxy server S2 to forward the information to the user at pseudonym P, proxy server S2 will refuse to honor the request, unless the message includes credentials for the accessor adequate to meet the requirements the user has chosen to impose, as above, on third parties who wish to send information to the user. If the message does include adequate credentials, then proxy server S2 removes a single-use pseudonymous return address envelope from it s database record for pseudonym P, and uses the envelope to send a message containing the specified information along a secure mix path to the user of pseudonym P. If the envelope being used is the only envelope stored for pseudonym P, or more generally if the supply of such envelopes is low, proxy server S2 adds a notation to this message before sending it, which notation indicates to the user's local server that it should send additional envelopes to proxy server S2 for future use.

In a more general variation, the user may instruct the proxy server S2 to impose more complex requirements on the granting of requests by third parties, not simply Boolean combinations of required credentials. The user may impose any Boolean combination of simple requirements that may include, but are not limited to, the following:

(a.) the accessor (third party) is a particular party

(b.) the accessor has provided a particular credential

(c.) satisfying the request would involve disclosure to the accessor of a certain fact about the user's user profile

(d.) satisfying the request would involve disclosure to the accessor of the user's target profile interest summary

(e.) satisfying the request would involve disclosure to the accessor of statistical summary data, which data are computed from the useres user profile or target profile interest summary together with the user profiles and target profile interest summaries of at least n other users in the user base of the proxy server

(f.) the content of the request is to send the user a target object, and this target object has a particular attribute (such as high reading level, or low vulgarity, or an authenticated Parental Guidance rating from the MPAA)

(g.) the content of the request is to send the user a target object, and this target object has been digitally signed with a particular private key (such as the private key used by the National Pharmaceutical Association to certify approved documents)

(h.) the content of the request is to send the user a target object, and the target profile has been digitally signed by a profile authentication agency, guaranteeing that the target profile is a true and accurate profile of the target object it claims to describe, with all attributes authenticated.

(i.) the content of the request is to send the user a target object, and the target profile of this target object is within a specified distance of a particular search profile specified by the user

(j.) the content of the request is to send the user a target object, and the proxy server S2,.by using the user's stored target profile interest summary, estimates the user's likely interest in the target object to be above a specified threshold

(k.) the accessor indicates its willingness to make a particular payment to the user in exchange for the fulfillment of the request

The steps required to create and maintain the user's access-control requirements are as follows:

1. The user composes a Boolean combination of predicates that apply to requests; the resulting complex predicate should be true when applied to a request that the user wants proxy server S2 to honor, and false otherwise. The complex predicate may be encoded in another form, for efficiency.
2. The complex predicate is signed with SK_P, and transmitted from the user's client processor C3 to the proxy server S2 through the mix path enclosed in a packet that also contains the user's pseudonym P.
3. The proxy server S2 receives the packet, verifies its authenticity using PK_P and stores the access control instructions specified in the packet as part of its database record for pseudonym P. The proxy server S2 enforces access control as follows:

1. The third party (accessor) transmits a request to proxy server S2 using the normal point-to-point connections provided by the network N. The request may be to access the target profile interest summaries associated with a set of pseudonyms P1 . . . Pn, or to access the user profiles associated with a set of pseudonyms P1 . . . Pn, or to forward a message to the users associated with pseudonyms P1 . . . Pn. The accessor may explicitly specify the pseudonyms P1 . . . Pn, or may ask that P1 . . . Pn be chosen to be the set of all pseudonyms registered with proxy server S2 that meet specified conditions.
2. The proxy server S2 indexes the database record for each pseudonym Pi (1<=I<=n), retrieves the access requirements provided by the user associated with Pi, and determines whether and how the transmitted request should be satisfied for Pi. If the requirements are satisfied, S2 proceeds with steps 3a–3c.

3a. If the request can be satisfied but only upon payment of a fee, the proxy server S2 transmits a payment request to the accessor, and waits for the accessor to send the payment to the proxy server S2. Proxy server S2 retains a service fee and forward s the balance of the payment to the user associated with pseudonym Pi, via an anonymous return packet that this user has provided.

3b. If the request can be satisfied but only upon provision of a credential, the proxy server S2 transmits a credential request to the accessor, and waits for the accessor to send the credential to the proxy server S2.

3c. The proxy server S2 satisfies the request by disclosing user-specific information to the accessor, by providing the accessor with a set of single-use envelopes to communicate directly with the user, or by forwarding a message to the user, as requested.

4. Proxy server S2 optionally sends a message to the accessor, indicating why each of the denied requests for P1 . . . Pn was denied, and/or indicating how many requests were satisfied.
5. The active and/or passive relevance feedback provided by any user U with respect to any target object sent by any path from the accessor is tabulated by the above-described tabulating process resident on user U's client processor C3. As described above, a summary of such

information is periodically transmitted to the proxy server S2 to enable the proxy server S2 to update that user's target profile interest summary and user profile.

The access control criteria can be applied to solicited as well as unsolicited transmissions. That is, the proxy server can be used to protect the user from inappropriate or misrepresented target objects that the user may request. If the user requests a target object from an information server, but the target object turns out not to meet the access control criteria, then the proxy server will not permit the information server to transmit the target object to the user, or to charge the user for such transmission. For example, to guard against target objects whose profiles have been tampered with, the user may specify an access control criterion that requires the provider to prove the target profile's accuracy by means of a digital signature from a profile authentication agency. As another example, the parents of a child user may instruct the proxy server that only target objects that have been digitally signed by a recognized child protection organization may be transmitted to the user; thus, the proxy server will not let the user retrieve pornography, even from a rogue information server that is willing to provide pornography to users who have not supplied an adulthood credential.

Distribution of Information with Multicast Trees

The graphical representation of the network N presented in FIG. 3 shows that at least one of the data communications links can be eliminated, as shown in FIG. 4, while still enabling the network N to transmit messages among all the servers A–D. By elimination, we mean that the link is unused in the logical design of the network, rather than a physical disconnection of the link. The graphs that result when all redundant data communications links are eliminated are termed "trees" or "connected acyclic graphs." A graph where a message could be transmitted by a server through other servers and then return to the transmitting server over a different originating data communications link is termed a "cycle." A tree is thus an acyclic graph whose edges (links) connect a set of graph "nodes" (servers). The tree can be used to efficiently broadcast any data file to selected servers in a set of interconnected servers.

The tree structure is attractive in a communications network because much information distribution is multicast in nature—that is, a piece of information available at a single source must be distributed to a multiplicity of points where the information can be accessed. This technique is widely known: for example, "FAX trees" are in common use in political organizations, and multicast trees are widely used in distribution of multimedia data in the Internet; for example, see "Scaleable Feedback Control for Multicast Video Distribution in the Internet," (Jean-Chrysostome Bolot, Thierry Turletti, & Ian Wakeman, Computer Communication Review, Vol. 24, #4, October, '94, Proceedings of SIGCOMM'94, pp. 58–67) or "An Architecture For Wide-Area Multicast Routing," (Stephen Deering, Deborah Estrin, Dino Farinacci, Van Jacobson, Ching-Gung Liu, & Liming Wei, Computer Communication Review, Vol. 24, #4, October, '94, Proceedings of SIGCOMM'94, pp. 126–135). While there are many possible trees that can be overlaid on a graph representation of a network, both the nature of the networks (e.g., the cost of transmitting data over a link) and their use (for example, certain nodes may exhibit more frequent intercommunication) can make one choice of tree better than another for use as a multicast tree. One of the most difficult problems in practical network design is the construction of "good" multicast trees, that is, tree choices which exhibit low cost (due to data not traversing links unnecessarily) and good performance (due to data frequently being close to where it is needed)

Constructing a Multicast Tree

Algorithms for constructing multicast trees have either been ad-hoc, as is the case of the Deering, et al. Internet multicast tree, which adds clients as they request service by grafting them into the existing tree, or by construction of a minimum cost spanning tree. A distributed algorithm for creating a spanning tree (defined as a tree that connects, or "spans," all nodes of the graph) on a set of Ethernet bridges was developed by Radia Perlman ("Interconnections: Bridges and Routers," Radia Perlman, Addison-Wesley, 1992). Creating a minimal-cost spanning tree for a graph depends on having a cost model for the arcs of the graph (corresponding to communications links in the communications network). In the case of Ethernet bridges, the default cost (more complicated costing models for path costs are discussed on pp. 72–73 of Perlman) is calculated as a simple distance measure to the root; thus the spanning tree minimizes the cost to the root by first electing a unique root and then constructing a spanning tree based on the distances from the root. In this algorithm, the root is elected by recourse to a numeric ID contained in "configuration messages": the server w hose ID has minimum numeric value is chosen as the root. Several problems exist with this algorithm in general. First, the method of using an ID does not necessarily select the best root for the nodes interconnected in the tree. Second, the cost model is simplistic.

We first show how to use the similarity-based methods described above to select the servers most interested in a group of target objects, herein termed "core servers" for that group. Next we show how to construct an unrooted multicast tree that can be used to broadcast files to these core servers. Finally, we show how files corresponding to target objects are actually broadcast through the multicast tree at the initiative of a client, and how these files are later retrieved from the core servers when clients request them.

Since the choice of core servers to distribute a file to depends on the set of users who are likely to retrieve the file (that is, the set of users who are likely to be interested in the corresponding target object), a separate set of core servers and hence a separate multicast tree may be used for each topical group of target objects. Throughout the description below, servers may communicate among themselves through any path over which messages can travel; the goal of each multicast tree is to optimize the multicast distribution of files corresponding to target objects of the corresponding topic. Note that this problem is completely distinct from selecting a multiplicity of spanning trees for the complete set of interconnected nodes as disclosed by Sincoskie in U.S. Pat. No. 4,706,080 and the publication titled "Extended Bridge Algorithms for Large Networks" by W. D. Sincoskie and C. J. Cotton, published January 1988 in IEEE Network on pages 16–24. The trees in this disclosure are intentionally designed to interconnect a selected subset of nodes in the system, and are successful to the degree that this subset is relatively small.

Multicast Tree Construction Procedure

A set of topical multicast trees for a set of homogenous target objects may be constructed or reconstructed at any time, as follows. The set of target objects is grouped into a fixed number of topical clusters C1 . . . Cp with the methods described above, for example, by choosing C1 . . . Cp to be the result of a k-means clustering of the set of target objects, or alternatively a covering set of low-level clusters from a hierarchical cluster tree of these target objects. A multicast tree MT(c) is then constructed from each cluster C in C1 . . . Cp, by the following procedure:

1. Given a set of proxy servers, S1 . . . Sn, and a topical cluster C. It is assumed that a general multicast tree MT_{full}

that contains all the proxy servers S1 . . . Sn has previously been constructed by well-known methods.

2. Each pair <Si, C> is associated with a weight, w(Si, C), which is intended to covary with the expected number of users in the user base of proxy server Si who will subsequently access a target object from cluster C. This weight is computed by proxy server Si in any of several ways, all of which make use of the similarity measurement computation described herein.

One variation makes use of the following steps: (a) Proxy server Si randomly selects a target object T from cluster C. (b) For each pseudonym in its local database, with associated user U, proxy server Si applies the techniques disclosed above to user U's stored user profile and target profile interest summary in order to estimate the interest w(U, T) that user U has in t he selected target object T. The aggregate interest w(Si, T) that the user base of proxy server Si has in the target object T is defined to be the sum of these interest values w(U, T). Alternatively, w(Si, T) may be defined to be the sum of values s(w(U, T)) over all U in the user base. Here s(*) is a sigmoidal function that is close to 0 for small arguments and close to a constant P_{max} for large arguments; thus s(w(U, T)) estimates the probability that user U will access target object T, which probability is assumed to be independent of the probability that any other user will access target object T. In a variation, w(Si, T) is made to estimate the probability that at least one user from the user base of Si will access target object T: then w(Si, T) may be defined as the maximum of values w(U, T), or of 1 minus the product over the users U of the quantity (1−s(w(U, T))). (c)Proxy server Si repeats steps (a)–(b) for several target objects T selected randomly from cluster C, and averages the several values of w(Si, T) thereby computed in step (b) to determine the desired quantity w(Si, C), which quantity represents the expected aggregate interest by the user base of proxy server Si in the target objects of cluster C.

In another variation, where target profile interest summaries are embodied as search profile sets, the following procedure is followed to compute w(Si, C): (a). For each search profile P_S in the locally stored search profile set of any user in the user base of proxy server Si, proxy server Si computes the distance $d(P_S, P_C)$ between the search profile and the cluster profile P_C of cluster C. (b). w(Si,C) is chosen to be the maximum value of $(-d(P_S,P_C)/r)$ across all such search profiles P_S, where r is computed as an affine function of the cluster diameter of cluster C. The slope and/or intercept of this affine function are chosen to be smaller (thereby increasing w(Si, C)) for servers Si for which the target object provider wishes to improve performance, as may be the case if the users in the user base of proxy server Si pay a premium for improved performance, or if performance at Si will otherwise be unacceptably low due to slow network connections.

In another variation, the proxy server Si is modified so that it maintains not only target profile interest summaries for each user in its user base, but also a single aggregate target profile interest summary for the entire user base. This aggregate target profile interest summary is determined in the usual way from relevance feedback, but the relevance feedback on a target object, in this case, is considered to be the frequency with which users in the user base retrieved the target object when it was new. Whenever a user retrieves a target object by means of a request to proxy server Si, the aggregate target profile interest summary for proxy server Si is updated. In this variation, w(Si, C) I s estimated by the following steps:

(a) Proxy server Si randomly selects a target object T from cluster C.

(b) Proxy server Si applies the techniques disclosed above to its stored aggregate target profile interest summary in order to estimate the aggregate interest w(Si, T) that its aggregated user base had in the selected target object T, when new; this may be interpreted as an estimate of the likelihood that at least one member of the user base will retrieve a new target object similar to T.

(c) Proxy server Si repeats steps (a)–(b) for several target objects T selected randomly from cluster C, and averages the several values of w(Si, T) thereby computed in step (b) to determine the desired quantity w(Si, C), which quantity represents the expected aggregate interest by the user base of proxy server Si in the target objects of cluster C.

3. Those servers Si from among S1 . . . Sn with the greatest weights w(Si, C) are designated "core servers" for cluster C. In one variation, where it is desired to select a fixed number of core servers, those servers Si with the greatest values of w(Si, C) are selected. In another variation, the value of w(Si, C) for each server Si is compared against a fixed threshold w_{min}, and those servers Si such that w(Si, C) equals or exceeds w_{min} are selected as core servers. If cluster C represents a narrow and specialized set of target objects, as often happens when the clusters C1 . . . Cp are numerous, it is usually adequate to select only a small number of core server cluster C, thereby obtaining substantial advantages in computational efficiency in steps 4–5 below

4. A complete graph G(C) is constructed whose vertices are the designated core servers for cluster C. For each pair of core servers, the cost of transmitting a message between those core servers along the cheapest path is estimated, and the weight of the edge connecting those core servers is taken to be this cost. The cost is determined as a suitable function of average transmission charges, average transmission delay, and worst-case or near-worst-case transmission delay.

5. The multicast tree MT(C) is computed by standard methods to be the minimum spanning tree (or a near-minimum spanning tree) for G(C), where the weight of an edge between two core servers is taken to be the cost of transmitting a message between those two core servers. Note that MT(C) does not contain as vertices all proxy servers S1 . . . Sn, but only the core servers for cluster C.

6. A message M is formed describing the cluster profile for cluster C, the core servers for cluster C and the topology of the multicast tree MT(C) constructed on those core servers. Message M is broadcast to all proxy servers S1 . . . Sn by means of the general multicast tree MT_{full}. Each proxy server Si, upon receipt of message M, extracts the cluster profile of cluster C, and stores it on a local storage device, together with certain other information that it determines from message M, as follows. If proxy server Si is named in message M as a core server for cluster C, then proxy server Si extracts and stores the subtree of MT(C) induced by all core servers whose path distance from Si in the graph MT(C) is less than or equal to d, where d is a constant positive integer (usually from 1 to 3). If message M does not name proxy server Si as a core server for MT(C), then proxy server Si extracts and stores a list of one or more nearby core servers that can be inexpensively contacted by proxy server Si over virtual point-to-point links.

In the network of FIG. 3, to illustrate the use of trees, as applied to the system of the present invention, consider the following simple example where it is assumed that client r provides on-line information for the network, such as an

electronic newspaper. This information can be structured by client r into a prearranged form, comprising a number of files, each of which is associated with a different target object. In the case of an electronic newspaper, the files can contain textual representations of stock prices, weather forecasts, editorials, etc. The system determines likely demand for the target objects associated with these files in order to optimize the distribution of the files through the network N of interconnected clients p–s and proxy servers A–D. Assume that cluster C consists of text articles relating to the aerospace industry; further assume that the target profile interest summaries stored at proxy servers A and B for the users at clients p and r indicate that these users are strongly interested in such articles. Then the proxy servers A and B are selected as core servers for the multicast tree MT(C). The multicast tree MT(C) is then computed to consist of the core servers, A and B, connected by an edge that represents the least costly virtual point-to-point link between A and B (either the direct path A-B or the indirect path A-C-B, depending on the cost).

Global Requests to Multicast Trees

One type of message that may be transmitted to any proxy server S is termed a "global request message." Such a message M triggers the broadcast of an embedded request R to all core servers in a multicast tree MT(C). The content of request R and the identity of cluster C are included in the message M, as is a field indicating that message M is a global request message. In addition, the message M contains a field S_{last} which is unspecified except under certain circumstances described below, when it names a specific core server. A global request message M may be transmitted to proxy server S by a user registered with proxy server S, which transmission may take place along a pseudonymous mix path, or it may be transmitted to proxy server S from another proxy server, along a virtual point-to-point connection.

When a proxy server S receives a message M that is marked as a global request message, it acts as follows: 1. If proxy server S is not a core server for topic C, it retrieves its locally stored list of nearby core servers for topic C, selects from this list a nearby core server S', and transmits a copy of message M over a virtual point-to-point connection to core server S'. If this transmission fails, proxy server S repeats the procedure with other core servers on its list. 2. If proxy server S is a core server for topic C, it executes the following steps: (a) Act on the request R that is embedded in message M. (b) Set S_{curr} to be S(C) Retrieve the locally stored subtree of MT(C), and extract from it a list L of all core servers that are directly linked to S_{curr} in this subtree. (d) If the message M specifies a value for S_{last} and S_{last} appears on the list L, remove S_{last} from the list L. Note that list L may be empty before this step, or may become empty as a result of this step. (e) For each server Si in list L, transmit a copy of message M from server S to server Si over a virtual point-to-point connection, where the S_{last} field of the copy of message M has been altered to S_{curr}. If Si cannot be reached in a reasonable amount of time by any virtual point-to-point connection (for example, server Si is broken), recurse to step (c) above with S_{orig} bound to S_{curr} and S_{curr} bound to S{\sub I} for the duration of the recursion.

When server S' in step 1 or a server Si in step 2(e) receives a copy of the global request message M, it acts according to exactly the same steps. As a result, all core servers eventually receive a copy of global request message M and act on the embedded request R, unless some core servers cannot be reached. Even if a core server is unreachable, step (e) ensures that the broadcast can continue to other core servers in most circumstances, provided that d>1; higher values of d provide additional insurance against unreachable core servers.

Multicastini Files

The system for customized electronic information of desirable objects executes the following steps in order to introduce a new target object into the system. These steps are initiated by an entity E, which may be either a user entering commands via a keyboard at a client processor q, as illustrated in FIG. **3**, or an automatic software process resident on a client or server processor q. 1. Processor q forms a signed request R, which asks the receiver to store a copy of a file F on its local storage device. File F, which is maintained by client q on storage at client q or on storage accessible by client q over the network, contains the informational content of or an identifying description of a target object, as described above. The request R also includes an address at which entity E may be contacted (possibly a pseudonymous address at some proxy server D), and asks the receiver to store the fact that file F is maintained by an entity at said address. 2. Processor q embeds request R in a message M1, which it pseudonymously transmits to the entity E's proxy server D as described above. Message M1 instructs proxy server D to broadcast request R along an appropriate multicast tree. 3. Upon receipt of message M1, proxy server D examines the doubly embedded file F and computes a target profile P for the corresponding target object. It compares the target profile P to each of the cluster profiles for topical clusters C1 . . . Cp described above, and chooses Ck to be the cluster with the smallest similarity distance to profile P. 4. Proxy server D sends itself a global request message M instructing itself to broadcast request R along the topical multicast tree MT(Ck). 5. Proxy server D notifies entity E through a pseudonymous communication that file F has been multicast along the topical multicast tree for cluster Ck.

As a result of the procedure that server D and other servers follow for acting on global request messages, step 4 eventually causes all core servers for topic Ck to act on request R and therefore store a local copy of file F. In order to make room for file F on its local storage device, a core server Si may have to delete a less useful file. There are several ways to choose a file to delete. One option, well known in the art, is for Si to choose to delete the least recently accessed file. In another variation, Si deletes a file that it believes few users will access. In this variation, whenever a server Si stores a copy of a file F, it also computes and stores the weight w(Si, C_F), where C_F is a cluster consisting of the single target object associated with file F. Then, when server Si needs to delete a file, it chooses to delete the file F with the lowest weight w(Si, C_F). To reflect the fact that files are accessed less as they age, server Si periodically multiplies its stored value of w(Si, C_F) by a decay factor, such as 0.95, for each file F that it then stores. Alter natively, instead of using a decay factor, server Si may periodically recompute aggregate interest w(Si, C_F) for each file F that it stores; the aggregate interest changes over time because target objects typically have an age attribute that the system considers in estimating user interest, as described above.

If entity E later wishes to remove file F from the network, for example because it has just multicast an updated version, it pseudonymously transmits a digitally signed global request message to proxy server D, requesting all proxy servers in the multicast tree MT(Ck) to delete any local copy of file F that they may be storing.

Queries to Multicast Trees

In addition to global request messages, another type of message that may be transmitted to any proxy server S is

termed a "query message." When transmitted to a proxy server, a query message causes a reply to be sent to the originator of the message; this reply will contain an answer to a given query Q if any of the servers in a given multicast tree MT(C) are able to answer it, and will otherwise indicate that no answer is available. The query and the cluster C are named in the query message. In addition, the query message contains a field S_{last} which is unspecified except under certain circumstances described below, when it names a specific core server. When a proxy server S receives a message M that is marked as a query message, it acts as follows:1. Proxy server S sets A_r to be the return address for the client or server that transmitted message M to server S. A_r may be either a network address or a pseudonymous address 2. If proxy server S is not a core server for cluster C, it retrieves its locally stored list of nearby core servers for topic C, selects from this list a nearby core server S', and transmits a copy of the locate message M over a virtual point-to-point connection to core server S'. If this transmission fails, proxy server S repeats the procedure with other core servers on its list. Upon receiving a reply, it forwards this reply to address A_r. 3. If proxy server S is a core server for cluster C, and it is able to answer query Q using locally stored information, then it transmits a "positive" reply to A_r containing the answer. 4. If proxy server S is a core server for topic C, but it is unable to answer query Q using locally stored information, then it carries out a parallel depth-first search by executing the following steps: (a) Set L to be the empty list. (b) Retrieve the locally stored subtree of MT(C). For each server Si directly linked to S_{curr} in this subtree, other than S_{last} (if specified), add the ordered pair (Si, S) to the list L. (c) If L is empty, transmit a "negative" reply to address A_r saying that server S cannot locate an answer to query Q, and terminate the execution of step 4; otherwise proceed to step (d). (d) Select a list L1 of one or more server pairs (Ai, Bi) from the list L. For each server pair (Ai, Bi) on the list L1, form a locate message M(Ai, Bi), which is a copy of message M whose S_{last} field has been modified to specify Bi, and transmit this message M(Ai, Bi) to server Ai over a virtual point-to-point connection. (e) For each reply received (by S) to a message sent in step (d), act as follows: (I) If a "positive" reply arrives to a locate message M(Ai, Bi), then forward this reply to A_r and terminate step 4, immediately. (ii) If a "negative" reply arrives to a locate message M(Ai, Bi), then remove the pair (Ai, Bi) from the list L1. (iii) If the message M(Ai, Bi) could not be successfully delivered to Ai, then remove the pair (Ai, Bi) from the list L1, and add the pair (Ci, Ai) to the list L1 for each Ci other than Bi that is directly linked to Ai in the locally stored subtree of MT(C). (f) Once L1 no longer contains any pair (Ai, Bi) for which a message M(Ai, Bi) has been sent, or after a fixed period of time has elapsed, return to step (c).

Retrieving Files from a Multicast Tree

When a processor q in the network wishes to retrieve the file associated with a given target object, it executes the following steps. These steps are initiated by an entity E, which may be either a user entering commands via a keyboard at a client q, as illustrated in FIG. 3, or an automatic software process resident on a client or server processor q. 1. Processor q forms a query Q that asks whether the recipient (a core server for cluster C) still stores a file F that was previously multicast to the multicast tree MT(C); if so, the recipient server should reply with its own server name. Note that processor q must already know the name of file F and the identity of cluster C; typically, this information is provided to entity E by a service such as the news clipping service or browsing system described below, which must identify files to the user by (name, multicast topic) pair. 2. Processor q forms a query message M that poses query Q to the multicast tree MT(C). 3. Processor q pseudonymously transmits message M to the user's proxy server D, as described above. 4. Processor q receives a response M2 to message M. 5. If the response M2 is "positive," that is, it names a server S that still stores file F, then processor q pseudonymously instructs the user's proxy server D to retrieve file F from server S. If the retrieval fails because server S has deleted file F since it answered the query, then client q returns to step 1. 6. If the response M2 is "negative," that is, it indicates that no server in MT(C) still stores file F, then processor q forms a query Q that asks the recipient for the address A of the entity that maintains file F; this entity will ordinarily maintain a copy of file F indefinitely. All core servers in MT(C) ordinarily retain this information (unless instructed to delete it by the maintaining entity), even if they delete file F for space reasons. Therefore, processor q should receive a response providing address A, whereupon processor q pseudonymously instructs the user's proxy server D to retrieve file F from address A.

When multiple versions of a file F exist on local servers throughout the data communication network N, but are not marked as alternate versions of the same file, the system's ability to rapidly locate files similar to F (by treating them as target objects and applying the methods disclosed in "Searching for Target Objects" above) makes it possible to find all the alternate versions, even if they are stored remotely. These related data files may then be reconciled by any method. In a simple instantiation, all versions of the data file would be replaced with the version that had the latest date or version number. In another instantiation, each version would be automatically annotated with references or pointers to the other versions.

NEWS CLIPPING SERVICE

The system for customized electronic identification of desirable objects of the present invention can be used in the electronic media system of FIG. 1 to implement an automatic news clipping service which learns to select (filter) news articles to match a user's interests, based solely on which articles the user chooses to read. The system for customized electronic identification of desirable objects generates a target profile for each article that enters the electronic media system, based on the relative frequency of occurrence of the words contained in the article. The system for customized electronic identification of desirable objects also generates a search profile set for each user, as a function of the target profiles of the articles the user has accessed and the relevance feedback the user has provided on these articles. As new articles are received for storage on the mass storage systems SS_1–SS_m of the information servers I_1–I_m, the system for customized electronic identification of desirable objects generates their target profiles. The generated target profiles are later compared to the search profiles in the users' search profile sets, and those new articles whose target profiles are closest (most similar) to the closest search profile in a user's search profile set are identified to that user for possible reading. The computer program providing the articles to the user monitors how much the user reads (the number of screens of data and the number of minutes spent reading), and adjusts the search profiles in the user's search profile set to more closely match what the user apparently prefers to read. The details of the method used by this system are disclosed in flow diagram form in FIG. 5. This method requires selecting a specific method of calculating user-

specific search profile sets, of measuring similarity between two profiles, and of updating a user's search profile set (or more generally target profile interest summary) based on what the user read, and the examples disclosed herein are examples of the many possible implementations that can be used and should not be construed to limit the scope of the system.

Initialize Users' Search Profile Sets

The news clipping service instantiates target profile interest summaries as search profile sets, so that a set of high-interest search profiles is stored for each user. The search profiles associated with a given user change over time. As in any application involving search profiles, they can be initially determined for a new user (or explicitly altered by an existing user) by any of a number of procedures, including the following preferred methods: (1) asking the user to specify search profiles directly by giving keywords and/or numeric attributes, (2) using copies of the profiles of target objects or target clusters that the user indicates are representative of his or her interest, (3) using a standard set of search profiles copied or otherwise determined from the search profile sets of people who are demographically similar to the user.

Retrieve New Articles from Article Source

Articles are available on-line from a wide variety of sources. In the preferred embodiment, one would use the current days news as supplied by a news source, such as the AP or Reuters news wire. These news articles are input to the electronic media system by being loaded into the mass storage system SS_4 of an information server S_4. The article profile module **201** of the system for customized electronic identification of desirable objects can reside on the information server S4 and operates pursuant to the steps illustrated in the flow diagram of FIG. **5**, where, as each article is received at step **501** by the information server S_4, the article profile module **201** at step **502** generates a target profile for the article and stores the target profile in an article indexing memory (typically part of mass storage system SS_4 for later use in selectively delivering articles to users. This method is equally useful for selecting which articles to read from electronic news groups and electronic bulletin boards, and can be used as part of a system for screening and organizing electronic mail ("e-mail").

Calculate Article Profiles

A target profile is computed for each new article, as described earlier. The most important attribute of the target profile is a textual attribute that stands for the entire text of the article. This textual attribute is represented as described earlier, as a vector of numbers, which numbers in the preferred embodiment include the relative frequencies (TF/IDF scores) of word occurrences in this article relative to other comparable articles. The server must count the frequency of occurrence of each word in the article in order to compute the TF/IDF scores.

These news articles are then hierarchically clustered in a hierarchical cluster tree at step **503**, which serves as a decision tree for determining which news articles are closest to the user's interest. The resulting clusters can be viewed as a tree in which the top of the tree includes all target objects and branches further down the tree represent divisions of the set of target objects into successively smaller subclusters of target objects. Each cluster has a cluster profile, so that at each node of the tree, the average target profile (centroid) of all target objects stored in the subtree rooted at that node is stored. This average of target profiles is computed over the representation of target profiles as vectors of numeric attributes, as described above.

Compare Current Articles' Target Profiles to a User's Search Profiles

The process by which a user employs this apparatus to retrieve news articles of interest is illustrated in flow diagram form in FIG. **11**. At step **1101**, the user logs into the data communication network N via their client processor C_1 and activates the news reading program. This is accomplished by the user establishing a pseudonymous data communications connection as described above to a proxy server S_2, which provides front-end access to the data communication network N. The proxy server S_2 maintains a list of authorized pseudonyms and their corresponding public keys and provides access and billing control. The user has a search profile set stored in the local data storage medium on the proxy server S_2. When the user requests access to "news" at step **1102**, the profile matching module **203** resident on proxy server S_2 sequentially considers each search profile p_k from the user's search profile set to determine which news articles are most likely of interest to the user. The news articles were automatically clustered into a hierarchical cluster tree at an earlier step so that the determination can be made rapidly for each user. The hierarchical cluster tree serves as a decision tree for determining which articles' target profiles are most similar to search profile p_k: the search for relevant articles begins at the top of the tree, and at each level of the tree the branch or branches are selected which have cluster profiles closest to p_k. This process is recursively executed until the leaves of the tree are reached, identifying individual articles of interest to the user, as described in the section "Searching for Target Objects" above.

A variation on this process exploits the fact that many users have similar interests. Rather than carry out steps 5–9 of the above process separately for each search profile of each user, it is possible to achieve added efficiency by carrying out these steps only once for each group of similar search profiles, thereby satisfying many users' needs at once. In this variation, the system begins by non-hierarchically clustering all the search profiles in the search profile sets of a large number of users. For each cluster k of search profiles, with cluster profile p_k, it uses the method described in the section "Searching for Target Objects" to locate articles with target profiles similar to p_k. Each located article is then identified as of interest to each user who has a search profile represented in cluster k of search profiles.

Notice that the above variation attempts to match clusters of search profiles with similar clusters of articles. Since this is a symmetrical problem, it may instead be given a symmetrical solution, as the following more general variation shows. At some point before the matching process commences, all the news articles to be considered are clustered into a hierarchical tree, termed the "target profile cluster tree," and the search profiles of all users to be considered are clustered into a second hierarchical tree, termed the "search profile cluster tree." The following steps serve to find all matches between individual target profiles from any target profile cluster tree and individual search profiles from any search profile cluster tree: 1. For each child subtree S of the root of the search profile cluster tree (or, let S be the entire search profile cluster tree if it contains only one search profile): 2. Compute the cluster profile P_S to be the average of all search profiles in subtree S 3. For each subcluster (child subtree) T of the root of the target profile cluster tree (or, let T be the entire target profile cluster tree if it contains only one target profile): 4. Compute the cluster profile P_T to be the average of all target profiles in subtree T 5. Calculate $d(P_S, P_T$, the distance between P_S and P_T 6. If

$d(P_S, P_T)<t$, a threshold, 7. If S contains only one search profile and T contains only one target profile, declare a match between that search profile and that target profile, 8. otherwise recurse to step 1 to find all matches between search profiles in tree S and target profiles in tree T.

The threshold used in step 6 is typically an affine function or other function of the greater of the cluster variances (or cluster diameters) of S and T. Whenever a match is declared between a search profile and a target profile, the target object that contributed the target profile is identified as being of interest to the user who contributed the search profile. Notice that the process can be applied even when the set of users to be considered or the set of target objects to be considered is very small. In the case of a single user, the process reduces to the method given for identifying articles of interest to a single user. In the case of a single target object, the process constitutes a method for identifying users to whom that target object is of interest.

Present List of Articles to User

Once the profile correlation step is completed for a selected user or group of users, at step **1104** the profile processing module **203** stores a list of the identified articles for presentation to each user. At a user's request, the profile processing system **203** retrieves the generated list of relevant articles and presents this list of titles of the selected articles to the user, who can then select at step **1105** any article for viewing. (If no titles are available, then the first sentence(s) of each article can be used.) The list of article titles is sorted according to the degree of similarity of the article's target profile to the most similar search profile in the user's search profile set. The resulting sorted list is either transmitted in real time to the user client processor C_1, if the user is present at their client processor C_1, or can be transmitted to a user's mailbox, resident on the user's client processor C_1 or stored within the server S_2 for later retrieval by the user; other methods of transmission include facsimile transmission of the printed list or telephone transmission by means of a text-to-speech system. The user can then transmit a request by computer, facsimile, or telephone to indicate which of the identified articles the user wishes to review, if any. The user can still access all articles in any information server S_4 to which the user has authorized access, however, those lower on the generated list are simply further from the user's interests, as determined by the user's search profile set. The server S_2 retrieves the article from the local data storage medium or from an information server S_4 and presents the article one screen at a time to the user's client processor C_1. The user can at any time select another article for reading or exit the process.

Monitor Which Articles Are Read

The user's search profile set generator **202** at step **1107** monitors which articles the user reads, keeping track of how many pages of text are viewed by the user, how much time is spent viewing the article, and whether all pages of the article were viewed. This information can be combined to measure the depth of the user's interest in the article, yielding a passive relevance feedback score, as described earlier. Although the exact details depend on the length and nature of the articles being searched, a typical formula might be: measure of article attractiveness=0.2 if the second page is accessed +0.2 if all pages are accessed +0.2 if more than 30 seconds was spent on the article +0.2 if more than one minute was spent on the article +0.2 if the minutes spent in the article are greater than half the number of pages.

The computed measure of article attractiveness can then be used as a weighting function to adjust the user's search profile set to thereby more accurately reflect the user's dynamically changing interests.

Update User Profiles

Updating of a user's generated search profile set can be done at step **1108** using the method described in copending U.S. patent application Ser. No. 08/346,425. When an article is read, the server S_2 shifts each search profile in the set slightly in the direction of the target profiles of those nearby articles for which the computed measure of article attractiveness was high. Given a search profile with attributes u_{ik} from a user's search profile set, and a set of J articles available with attributes d_{jk} (assumed correct for now), where I indexes users, j indexes articles, and k indexes attributes, user I would be predicted to pick a set of P distinct articles to minimize the sum of $d(u_I, b_j)$ over the chosen articles j. The user's desired attributes u_{ik} and an article's attributes d_{jk} would be some form of word frequencies such as TF/IDF and potentially other attributes such as the source, reading level, and length of the article, while $d(u_I, d_j)$ is the distance between these two attribute vectors (profiles) using the similarity measure described above. If the user picks a different set of P articles than was predicted, the user search profile set generation module should try to adjust u and/or d to more accurately predict the articles the user selected. In particular, u_I and/or d_j should be shifted to increase their similarity if user I was predicted not to select article j but did select it, and perhaps also to decrease their similarity if user I was predicted to select article j but did not. A preferred method is to shift u for each wrong prediction that user I will not select article j, using the formula: $u_{ik}'=u_{ik}-e(u_{ik}\ d_{jk})$

Here u_I is chosen to be the search profile from user I's search profile set that is closest to target profile. If e is positive, this adjustment increases the match between user I's search profile set and the target profiles of the articles user I actually selects, by making u_I closer to d_j for the case where the algorithm failed to predict an article that the viewer selected. The size of e determines how many example articles one must see to change the search profile substantially. If e is too large, the algorithm becomes unstable, but for sufficiently small e, it drives u to its correct value. In general, e should be proportional to the measure of article attractiveness; for example, it should be relatively high if user I spends a long time reading article j. One could in theory also use the above formula to decrease the match in the case where the algorithm predicted an article that the user did not read, by making e negative in that case. However, there is no guarantee that u will move in the correct direction in that case. One can also shift the attribute weights w_I of user I by using a similar algorithm: $w_{ik}'=(w_{ik}-e|u_{ik-djk}|)/S_k\ (w_{ik}-e|u_{ik}-d_{jk}|)$ This is particularly important if one is combining word frequencies with other attributes. As before, this increases the match if e is positive—for the case where the algorithm failed to predict an article that the user read, this time by decreasing the weights on those characteristics for which the user's target profile u_I differs from the article's profile d_j. Again, the size of e determines how many example articles one must see to replace what was originally believed. Unlike the procedure for adjusting u, one also make use of the fact that the above algorithm decreases the match if e is negative—for the case where the algorithm predicted an article that the user did not read. The denominator of the expression prevents weights from shrinking to zero over time by renormalizing the modified weights w_I' so that they sum to one. Both u and w can be adjusted for each article accessed. When e is small, as it should be, there is no conflict between the two parts of the algorithm. The selected user's search profile set is updated at step **1108**.

Further Applications of the Filtering Technology

The news clipping service may deliver news articles (or advertisements and coupons for purchasables) to off-line

users as well as to users who are on-line. Although the off-line users may have no way of providing relevance feedback, the user profile of an off-line user U may be similar to the profiles of on-line users, for example because user U is demographically similar to these other users, and the level of user U's interest in particular target objects can therefore be estimated via the general interest-estimation methods described earlier. In one application, the news clipping service chooses a set of news articles (respectively, advertisements and coupons) that are predicted to be of interest to user U, thereby determining the content of a customized newspaper (respectively, advertising/coupon circular) that may be printed and physically sent to user U via other methods. In general, the target objects included in the printed document delivered to user U are those with the highest median predicted interest among a group G of users, where group G consists of either the single off-line user U, a set of off-line users who are demographically similar to user U, or a set of off-line users who are in the same geographic area and thus on the same newspaper delivery route. In a variation, user group G is clustered into several subgroups G1 . . . Gk; an average user profile Pi is created from each subgroup Gi; for each article T and each user profile Pi, the interest in T by a hypothetical user with user profile Pi is predicted, and the interest of article T to group G is taken to be the maximum interest in article T by any of these k hypothetical users; finally, the customized newspaper for user group G is constructed from those articles of greatest interest to group G.

The filtering technology of the news clipping service is not limited to news articles provided by a single source, but may be extended to articles or target objects collected from any number of sources. For example, rather than identifying new news articles of interest, the technology may identify new or updated World Wide Web pages of interest. In a second application, termed "broadcast clipping," where individual users desire to broadcast messages to all interested users, the pool of news articles is replaced by a pool of messages to be broadcast, and these messages are sent to the broadcast-clipping-service subscribers most interested in them. In a third application, the system scans the transcripts of all real-time spoken or written discussions on the network that are currently in progress and designated as public, and employs the news-clipping technology to rapidly identify discussions that the user may be interested in joining, or to rapidly identify and notify users who may be interested in joining an ongoing discussion. In a fourth application, the system scans the transcripts of all real time spoken, written or acoustic (e.g., audio or video streaming data) on the network that are currently in progress, and employs news clipping technology to rapidly identify content which is most appropriate for a particular advertisement or promotion that may pertain to the target object profile of the content presently occurring. In a fifth application, the method is used as a post-process that filters and ranks in order of interest the many target objects found by a conventional database search, such as a search for all homes selling for under $200,000 in a given area, for all 1994 news articles about Marcia Clark, or for all Italian-language films. In a sixth application, the method is used to filter and rank the links in a hypertext document by estimating the user's interest in the document or other object associated with each link. In a seventh application, paying advertisers, who may be companies or individuals, are the source of advertisements or other messages, which take the place of the news articles in the news clipping service. A consumer who buys a product is deemed to have provided positive relevance feedback on advertisements for that product, and a consumer who buys a product apparently because of a particular advertisement (for example, by using a coupon clipped from that advertisement) is deemed to have provided particularly high relevance feedback on that advertisement. Such feedback may be communicated to a proxy server by the consumer's client processor (if the consumer is making the purchase electronically), by the retail vendor, or by the credit-card reader (at the vendor's establishment) that the consumer uses to pay for the purchase. Given a database of such relevance feedback, the disclosed technology is then used to match advertisements with those users who are most interested in them; advertisements selected for a user are presented to that user by any one of several means, including electronic mail, automatic display on the users screen, or printing them on a printer at a retail establishment where the consumer is paying for a purchase. The threshold distance used to identify interest may be increased for a particular advertisement, causing the system to present that advertisement to more users, in accordance with the amount that the advertiser is willing to pay.

A further use of the capabilities of this system is to manage a user's investment portfolio. Instead of recommending articles to the user, the system recommends target objects that are investments. As illustrated above by the example of stock market investments, many different attributes can be used together to profile each investment. The user's past investment behavior is characterized in the user's search profile set or target profile interest summary, and this information is used to match the user with stock opportunities (target objects) similar in nature to past investments. The rapid profiling method described above may be used to determine a rough set of preferences for new users. Quality attributes used in this system can include negatively weighted attributes, such as a measurement of fluctuations in dividends historically paid by the investment, a quality attribute that would have a strongly negative weight for a conservative investor dependent on a regular flow of investment income. Furthermore, the user can set filter parameters so that the system can monitor stock prices and automatically take certain actions, such as placing buy or sell orders, or e-mailing or paging the user with a notification, when certain stock performance characteristics are met. Thus, the system can immediately notify the user when a selected stock reaches a predetermined price, without the user having to monitor the stock market activity. The user's investments can be profiled in part by a "type of investment" attribute (to be used in conjunction with other attributes), which distinguishes among bonds, mutual funds, growth stocks, income stocks, etc., to thereby segment the user's portfolio according to investment type. Each investment type can then be managed to identify investment opportunities and the user can identify the desired ratio of investment capital for each type, e.g., in accordance with the system's automatic recommendation for relative distribution of investment capital as indicated by the relative level of user interest for each type.

In one application the system may also keep track of the most relevant articles for the user who may receive recommendations also through notification (or paging for new releases). In the previously described preferred implementation, the similarity of articles was described in terms of the tendency of metrically similar users to read them where metric similarity of users is determined by the tendency of those users to read similar articles wherein feedback from all of the users is considered. In this application however, only those articles which tend to be read by

similar users which have a similar stock portfolio to that of the user are instead considered similar. Accordingly, owners of stocks which are metrically similar to certain articles are targeted with those articles. By applying similar techniques in this application to those herein described, relevance feedback determines the metric similarity of the associative attributes which is each stock, with the relevant associative attributes which are each article (or their associated textual, descriptive or numeric attributes contained therein). Additionally in this regard, it is also possible to bias the weighting values of users providing relevance feedback to favor those who have invested in similar types of stocks and who have a proven track record of success through their trading decisions. Another application for which this type of pre-adjusted relevance feedback is useful in recommending and/or automatically trading the most interesting stocks to users using the present methods above described, however, again biasing the relevance feedback to the system by those users who had been most successful in their past trading decisions with regards to those particular types of stocks. Because financial advisors possess varying degrees f skill which varies within different types of investments, such a collaborative filtering based market for investment need not be limited to stocks but to other types of investments as well. The market price for which this "expert advice" is purchased by would be investors, which have an infinity to investments of the particular types that those advisor are experts in may measured using the presently described techniques for determination of price point thus advice by a given expert for investments which had demonstrated a given level of success may be priced similarly. Additionally, some gross level feedback suggesting the advisors current awareness about investment types could be automatically assessed by passively observed which articles within which investment domains the user had been recently reading on-line. In accordance with the similarity techniques previously described, the user may browse between the genres of articles and stocks which are most relevant to one another. Because there are numerous systems and software tools which are used in attempting to predict both selected stocks and optimal times to buy or trade them, the current user customization techniques are best implemented as an enhancement feature to provide the user with not only quality but also personalization.

In the preferred implementation for an on-line newspaper or news filter, each of the above capabilities for customized recommendation and notification of investment related articles, stock recommendations and automated stock monitoring and trading features are provided to the user as an integrated financial news and investment service. Additionally, in accordance with the virtual communities section below described, users sharing common portfolios may wish to correspond on-line to advice or experiences with other similar users. Additionally, users who have a past track record of success may also be particularly identifiable through these virtual communities in conjunction with their participation or their comments and advice relating to specific stocks may be ascribed to those stocks, credentialed as originating from an expert with a proven track record (and made publicly available).

OTHER ON-LINE NEWSPAPER INTERFACE FEATURES

In accordance with current on-line news interface features, several implementation features of the present system include the following:

1. Automatically create a "customized newspaper". User profiling enabling custom recommendations may be achieved by purely passive means of user activity data or if desired, it can refine and automate the selection process of articles within user selected categories of interest as well as recommend articles within different categories which the user is likely to prefer as evidenced through past behaviors. Applications include:
   (a) Presentation of new articles and corresponding advertisements which are of highest interest to the user.
   (b) Recommending (highlighting) these articles from the directory.
2. A customized search engine which offers search results which are tailored and relevancy ranked to user preferences.
3. Using a survey for off-line users for subsequent issues, an inserted card inserted into each issue identifies or prioritizes the most interesting articles/ads.

Update Notification

A very important and novel characteristic of the architecture is the ability to identify new or updated target objects that are relevant to the user, as determined by the user's search profile set or target profile interest summary. ("Updated target objects" include revised versions of documents and new models of purchasable goods.) The system may notify the user of these relevant target objects by an electronic notification such as an e-mail message or facsimile transmission. In the variation where the system sends an e-mail message, the user's e-mail filter can then respond appropriately to the notification, for instance, by bringing the notification immediately to the user's personal attention, or by automatically submitting an electronic request to purchase the target object named in the notification. A simple example of the latter response is for the e-mail filter to retrieve an on-line document at a nominal or zero charge, or request to buy a purchasable of limited quantity such as a used product or an auctionable.

ACTIVE NAVIGATION (BROWSING)

Browsing by Navizating Through a Cluster Tree

A hierarchical cluster tree imposes a useful organization on a collection of target objects. The tree is of direct use to a user who wishes to browse through all the target objects in the tree. Such a user may be exploring the collection with or without a well-specified goal. The tree's division of target objects into coherent clusters provides an efficient method whereby the user can locate a target object of interest. The user first chooses one of the highest level (largest) clusters from a menu, and is presented with a menu listing the subclusters of said cluster, whereupon the user may select one of these subclusters. The system locates the subclusters, via the appropriate pointer that was stored with the larger cluster, and allows the user to select one of its subclusters from another menu. This process is repeated until the user comes to a leaf of the tree, which yields the details of an actual target object. Hierarchical trees allow rapid selection of one target object from a large set. In ten menu selections from menus of ten items (subclusters) each, one can reach 10^{10}=10,000,000,000 (ten billion) items. In the preferred embodiment, the user views the menus on a computer screen or terminal screen and selects from them with a keyboard or mouse. However, the user may also make selections over the telephone, with a voice synthesizer reading the menus and the user selecting subclusters via the telephone's touch-tone keypad. In another variation, the user simultaneously maintains two connections to the server, a telephone voice connection and a fax connection; the server sends successive menus to the user by fax, while the user selects choices via the telephone's touch-tone keypad.

Just as user profiles commonly include an associative attribute indicating the user's degree of interest in each target object, it is useful to augment user profiles with an additional associative attribute indicating the user's degree of interest in each cluster in the hierarchical cluster tree. This degree of interest may be estimated numerically as the number of subclusters or target objects the user has selected from menus associated with the given cluster or its subclusters, expressed as a proportion of the total number of subclusters or target objects the user has selected. This associative attribute is particularly valuable if the hierarchical tree was built using "soft" or "fuzzy" clustering, which allows a subclusters or target object to appear in multiple clusters: if a target document appears in both the "sports" and the "humor" clusters, and the user selects it from a menu associated with the "humor" cluster, then the system increases its association between the user and the "humor" cluster but not its association between the user and the "sports" cluster.

Labeling Clusters

Since a user who is navigating the cluster tree is repeatedly expected to select one of several subclusters from a menu, these subclusters must be usefully labeled (at step **503**), in such a way as to suggest their content to the human user. It is straightforward to include some basic information about each subcluster in its label, such as the number of target objects the subcluster contains (possibly just 1) and the number of these that have been added or updated recently. However, it is also necessary to display additional information that indicates the cluster's content. This content-descriptive information may be provided by a human, particularly for large or frequently accessed clusters, but it may also be generated automatically. The basic automatic technique is simply to display the cluster's "characteristic value" for each of a few highly weighted attributes. With numeric attributes, this may be taken to mean the cluster's average value for that attribute: thus, if the "year of release" attribute is highly weighted in predicting which movies a user will like, then it is useful to display average year of release as part of each cluster's label. Thus the user sees that one cluster consists of movies that were released around 1962, while another consists of movies from around 1982. For short textual attributes, such as "title of movie" or "title of document," the system can display the attribute's value for the cluster member (target object) whose profile is most similar to the cluster's profile (the mean profile for all members of the cluster), for example, the title of the most typical movie in the cluster. For longer textual attributes, a useful technique is to select those terms for which the amount by which the term's average TF/IDF score across members of the cluster exceeds the term's average TF/IDF score across all tar get objects is greatest, either in absolute terms or else as a fraction of the standard deviation of the term's TF/IDF score across all target objects. The selected terms are replaced with their morphological stems, eliminating duplicates (so that if both "slept" and "sleeping" were selected, they would be replaced by the single term "sleep") and optionally eliminating close synonyms or collocates (so that if both "nurse" and "medical" were selected, they might both be replaced by a single term such as "nurse," "medical," "medicine," or "hospital"). The resulting set of terms is displayed as part of the label. Finally, if freely redistributable thumbnail photographs or other graphical images are associated with some of the target objects in the cluster for labeling purposes, then the system can display as part of the label the image or images whose associated target objects have target profiles most similar to the cluster profile.

Users' navigational patterns may provide some useful feedback as to the quality of the labels. In particular, if users often select a particular cluster to explore, but then quickly backtrack and try a different cluster, this may signal that the first cluster's label is misleading. Insofar as other terms and attributes can pro video "next-best" alternative labels for the first cluster, such "next-best" labels can be automatically substituted for the misleading label. In addition, any user can locally relabel a cluster for his or her own convenience. Although a cluster label provided by a user is in general visible only to that user, it is possible to make global use of these labels via a "user labels" textual attribute for target objects, which attribute is defined for a given target object to be the concatenation of all labels provided by any user for any cluster containing that target object. This attribute influences similarity judgments: for example, it may induce the system to regard target articles in a cluster often labeled "Sports News" by users as being mildly similar to articles in an otherwise dissimilar cluster often labeled "International News" by users, precisely because the "user labels" attribute in each cluster profile is strongly associated with the term "News." The "user label" attribute is also used in the automatic generation of labels, just as other textual attributes are, so that if the user-generated labels for a cluster often include "Sports," the term "Sports" may be included in the automatically generated label as well.

It is not necessary for menus to be displayed as simple lists of labeled options; it is possible to display or print a menu in a form that shows in more detail the relation of the different menu options to each other. Thus, in a variation, the menu options are visually laid out in two dimensions or in a perspective drawing of three dimensions. Each option is displayed or printed as a textual or graphical label. The physical coordinates at which the options are displayed or printed are generated by the following sequence of steps: (1) construct for each option the cluster profile of the cluster it represents, (2) construct from each cluster profile its decomposition into a numeric vector, as described above, (3) apply singular value decomposition (SVD) to determine the set of two or three orthogonal linear axes along which these numeric vectors are most greatly differentiated, and (4) take the coordinates of each option to be the projected coordinates of that option's numeric vector along said axes. Step (3) may be varied to determine a set of, say, 6 axes, so that step (4) lays out the options in a 6-dimensional space; in this case the user may view the geometric projection of the 6-dimensional layout onto any plane passing through the origin, and may rotate this viewing plane in order to see differing configurations of the options, which emphasize similarity with respect to differing attributes in the profiles of the associated clusters. In the visual representation, the sizes of the cluster labels can be varied according to the number of objects contained in the corresponding clusters. In a further variation, all options from the parent menu are displayed in some number of dimensions, as just described, but with the option corresponding to the current menu replaced by a more prominent subdisplay of the options on the current menu; optionally, the scale of this composite display may be gradually increased over time, thereby increasing the area of the screen devoted to showing the options on the current menu, and giving the visual impression that the user is regarding the parent cluster and "zooming in" on the current cluster and its subclusters.

Further Navigational

It should be appreciated that a hierarchical cluster-tree may be configured with multiple cluster selections branching from each node or the same labeled clusters presented in

the form of single branches for multiple nodes ordered in a hierarchy. In one variation, the user is able to perform lateral navigation between neighboring clusters as well, by requesting that the system search for a cluster whose cluster profile resembles the cluster profile of the currently selected cluster. If this type of navigation is performed at the level of individual objects (leaf ends), then automatic hyperlinks may be then created as navigation occurs. This is one way that nearest neighbor clustering navigation may be performed. For example, in a domain where target objects are home pages on the World Wide Web, a collection of such pages could be laterally linked to create a "virtual mall". Most importantly, links to sites in the form of targeted advertisements may be temporarily generated (as a result of the user profile and the target object profile of the page being visited, the dialogue being conducted or the content being viewed, listened to or read at that moment). This is one way in which "on the fly" automatic creation of customized links may occur (user specific linking of advertisers with sites or other content including programming or joint ads or promotions between advertisers may occur in real time). Or in another period this technique may be used to recommend the most befitting sites and/or ads which should be linked together (based upon their similarity). Of course, certain promotions for example may be directly competitive such as a product for two brands of toothpaste. Such direct competitive overlap must thus be accounted for. This technique may also account for one way or two way (exchanged) links between vendors. Advertisers which exchange links or wish to link to a "prime location" should pay a price which is directly in accordance with the market demand for that advertisement though not exceeding the price value necessary to fill the available ad space. The techniques described in co-pending patent application entitled "PPS" suggests a method of automatically generating a customized motion (or joint promotion) for individual users. A similar technique may be used to automatically establish a price for the ad space (based on a combined predicted price per impression and predicted value for the average customer expected to access that advertisement. As feedback occurs, this pricing model is adjusted according to actual response feedback, links may be broken, reformed in a one way or two way context in automatic fashion as such.

The simplest way to use the automatic menuing system described above is for the user to begin browsing at the top of the tree and moving to more specific subclusters. However, in a variation, the user optionally provides a query consisting of textual and/or other attributes, from which query the system constructs a profile in the manner described herein, optionally altering textual attributes as described herein before decomposing them into numeric attributes. Query profiles are similar to the search profiles in a user's search profile set, except that their attributes are explicitly specified by a user, most often for one-time usage, and unlike search profiles, they are not automatically updated to reflect changing interests. A typical query in the domain of text articles might have "Tell me about the relation between Galileo and the Medici family" as the value of its "text of article" attribute, and 8 as the value of its "reading difficulty" attribute (that is, 8th-grade level). The system uses the method of section "Searching for Target Objects" above to automatically locate a small set of one or more clusters with profiles similar to the query profile, for example, the articles they contain are written at roughly an 8th-grade level and tend to mention Galileo and the Medicis. The user may start browsing at any of these clusters, and can move from it to subclusters, superclusters, and other nearby clusters. For a user who is looking for something in particular, it is generally less efficient to start at the largest cluster and repeatedly select smaller subclusters than it is to write a brief description of what one is looking for and then to move to nearby clusters if the objects initially recommended are not precisely those desired.

Although it is customary in information retrieval systems to match a query to a document, an interesting variation is possible where a query is matched to an already answered question. The relevant domain is a customer service center, electronic newsgroup, or Better Business Bureau where questions are frequently answered. Each new question-answer pair is recorded for future reference as a target object, with a textual attribute that specifies the question together with the answer provided. As explained earlier with reference to document titles, the question should be weighted more heavily than the answer when this textual attribute is decomposed into TF/IDF scores. A query specifying "Tell me about the relation between Galileo and the Medici family" as t he value of this attribute therefore locates a cluster of similar questions together with their answers. In a variation, each question-answer pair may be profiled with two separate textual attributes, one for the question and one for the answer. A query might then locate a cluster by specifying only the question attribute, or for completeness, both the question attribute and the (lower-weighted) answer attribute, to be the text "Tell me about the relation between Galileo and the Medici family."

The filtering technology described earlier can also aid the user in navigating among the target objects. When the system presents the user with a menu of subclusters of a cluster C of target objects, it can simultaneously present an additional menu of the most interesting target objects in cluster C, so that the user has the choice of accessing a subcluster or directly accessing one of the target objects. If this additional menu lists n target objects, then for each I between 1 and n inclusive, in increasing order, the I^{th} most prominent choice on this additional menu, which choice is denoted Top(C,i), is found by considering all target objects in cluster C that are further than a threshold distance t from all of Top(C,1), Top(C,2), . . . Top(C, I−1), and selecting the one in which the user's interest is estimated to be highest. If the threshold distance t is 0, then the menu resulting from this procedure simply displays the n most interesting objects in cluster C, but the threshold distance may be increased to achieve more variety in the target objects displayed. Generally the threshold distance t is chosen to be an affine function or other function of the cluster variance or cluster diameter of the cluster C.

As a novelty feature, the user U can "masquerade" as another user V, such as a prominent intellectual or a celebrity supernodel; as long as user U is masquerading as user V, the filtering technology will recommend articles not according to user U's preferences, but rather according to user V's preferences. Provided that user U has access to the user-specific data of user V, for example because user V has leased these data to user U for a financial consideration, then user U can masquerade as user V by instructing user U's proxy server S to temporarily substitute user V's user profile and target profile interest summary for user U's. In a variation, user U has access to an average user profile and an composite target profile interest summary for a group G of users; by instructing proxy server S to substitute these for user U's user-specific data, user U can masquerade as a typical member of group G, as is useful in exploring group preferences for sociological, political, or market research. More generally, user U may "partially masquerade" as

another user V or group G, by instructing proxy server S to temporarily replace user U's user-specific data with a weighted average of user U's user-specific data and the user-specific data for user V and group G.

Menu Organization

Although the topology of a hierarchical cluster tree is fixed by the techniques that build the tree, the hierarchical menu presented to the user for the user's navigation need not be exactly isomorphic to the cluster tree. The menu is typically a somewhat modified version of the cluster tree, reorganized manually or automatically so that the clusters most interesting to a user are easily accessible by the user. In order to automatically reorganize the menu in a user-specific way, the system first attempts automatically to identify existing clusters that are of interest to the user. The system may identify a cluster as interesting because the user often accesses target objects in that cluster—or, in a more sophisticated variation, because the user is predicted to have high interest in the cluster's profile, using the methods disclosed herein for estimating interest from relevance feedback.

Several techniques can then be used to make interesting clusters more easily accessible. The system can at the user's request or at all times display a special list of the most interesting clusters, or the most interesting subclusters of the current cluster, so that the user can select one of these clusters based on its label and jump directly to it. In general, when the system constructs a list of interesting clusters in this way, the I^{th} most prominent choice on the list, which choice is denoted Top(I), is found by considering all appropriate clusters C that are fairther than a threshold distance t from all of Top(1), Top(2), . . . Top(I-1), and selecting the one in which the user's interest is estimated to be highest. Here the threshold distance t is optionally dependent on the computed cluster variance or cluster diameter of the profiles in the latter cluster. Several techniques that reorganize the hierarchical menu tree are also usefil. First, menus can be reorganized so that the most interesting subcluster choices appear earliest on the menu, or are visually marked as interesting; for example, their labels are displayed in a special color or type face, or are displayed together with a number or graphical image indicating the likely level of interest. Second, interesting clusters can be moved to menus higher in the tree, i.e., closer to the root of the tree, so that they are easier to access if the user starts browsing at the root of the tree. Third, uninteresting clusters can be moved to menus lower in the tree, to make room for interesting clusters that are being moved higher. Fourth, clusters with an especially low interest score (representing active dislike) can simply be suppressed from the menus; thus, a user with children may assign an extremely negative weight to the "vulgarity" attribute in the determination of q, so that vulgar clusters and documents will not be available at all. As the interesting clusters and the documents in them migrate toward the top of the tree, a customized tree develops that can be more efficiently navigated by the particular user. If menus are chosen so that each menu item is chosen with approximately equal probability, then the expected number of choices the user has to make is minimized. If, for example, a user frequently accessed target objects whose profiles resembled the cluster profile of cluster (a, b, d) in FIG. **8** then the menu in FIG. **9** could be modified to show the structure illustrated in FIG. **10**.

In the variation where the general techniques disclosed herein for estimating a user's interest from relevance feedback are used to identify interesting clusters, it is possible for a user U to supply "temporary relevance feedback" to indicate a temporary interest that is added to his or her usual interests. This is done by entering a query as described above, i.e., a set of textual and other attributes that closely match the user's interests of the moment . This query becomes "active," and affects the system's determination of interest in either of two ways. In one approach, an active query is treated as if it were any other target object, and by virtue of being a query, it is taken to have received relevance feedback that indicates especially high interest. In an alternative approach, target objects X whose target profiles are similar to an active query's profile are simply considered to have higher quality q(U, X), in that q(U, X) is incremented by a term that increases with target object X's similarity to the query profile. Either strategy affects the usual interest estimates: clusters that match user U's usual interests (and have high quality q(*)) are still considered to be of interest, and clusters w hose profiles are similar to an active query are adjudged to have especially high interest. Clusters that are similar to both the query and the user's usual interests are most interesting of all. The user may modify or deactivate an active query at any time while browsing. In addition, if the user discovers a target object or cluster X of particular interest while browsing, he or she may replace or augment the original (perhaps vague) query profile with the target profile of target object or cluster X, t hereby amplifying or refining the original query to indicate an particular interest in objects similar to X. For example, suppose the user is browsing through documents, and specifies an initial query containing the word "Lloyd's," so that the system predicts documents containing the word "Lloyd's" to be more interesting and makes them more easily accessible, even to the point of listing such documents or clusters of such documents, as described above. In particular, certain articles about insurance containing the phrase "Lloyd's of London" are made more easily accessible, as are certain pieces of Welsh fiction containing phrases like "Lloyd's father." The user browses while this query is active, and hits upon a useful article describing the relation of Lloyd's of London to other British insurance houses; by replacing or augmenting the query with the full text of this article, the user can turn the attention of the system to other documents that resemble this article, such as documents about British insurance houses, rather than Welsh folk tales.

In a system where queries are used, it is useful to include in the target profiles an associative attribute that records the associations between a target object and whatever terms are employed in queries used to find that target object. The association score of target object X with a particular query term T is defined to be the mean relevance feedback on target object X, averaged over just those accesses of target object X that were made while a query containing term T was active, multiplied by the negated logarithm of term T's global frequency in all queries. The effect of this associative attribute is to increase the measured similarity of two documents if they are good responses to queries that contain the same terms. A further maneuver can be used to improve the accuracy of responses to a query: in the summation used to determine the quality q(U, X) of a target object X, a term is included that is proportional to the sum of association scores between target object X and each term in the active query, if any, so that target objects that are closely associated with terms in an active query are determined to have higher quality and therefore higher interest for the user. To complement the system's automatic reorganization of the hierarchical cluster tree, the user can be given the ability to reorganize the tree manually, as he or she sees fit. Any changes are optionally saved on the user's local storage

device so that they will affect the presentation of the tree in future sessions. For example, the user can choose to move or copy menu options to other menus, so that useful clusters can thereafter be chosen directly from the root menu of the tree or from other easily accessed or topically appropriate menus. In an other example, the user can select clusters C_1, C_2, . . . C_k listed on a particular menu M and choose to remove these clusters from the menu, replacing them on the menu with a single aggregate cluster M' containing all the target objects from clusters C_1, C_2, . . . C_k. In this case, the immediate subclusters of new cluster M' are either taken to be clusters C_1, C_2, . . . C_k themselves, or else, in a variation similar to the "scatter-gather" method, are automatically computed by clustering the set of all the subclusters of clusters C_1, C_2, . . . C_k according to the similarity of the cluster profiles of these subclusters.

Electronic Mall

In one application, the browsing techniques described above may be applied to a domain where the target objects are purchasable goods. When shoppers look for goods to purchase over the Internet or other electronic media, it is typically necessary to display thousands or tens of thousands of products in a fashion that helps consumers find the items they are looking for. The current practice is to use hand-crafted menus and sub-menus in which similar items are grouped together. It is possible to use the automated clustering and browsing methods described above to more effectively group and present the items. Purchasable items can be hierarchically clustered using a plurality of different criteria. Useful attributes for a purchasable item include but are not limited to a textual description and predefined category labels (if available), the unit price of the item, and an associative attribute listing the users who have bought this item in the past. Also useful is an associative attribute indicating which other items are often bought on the same shopping "trip" as this item; items that are often bought on the same trip will be judged similar with respect to this attribute, so tend to be grouped together. Retailers may be interested in utilizing a similar technique for purposes of predicting both the nature and relative quantity of items which are likely to be popular to their particular clientele. This prediction may be made by using aggregate purchasing records as the search profile set from which a collection of target objects is recommended. Estimated customer demand which is indicative of (relative) inventory quantity for each target object item is determined by measuring the cluster variance of that item compared to another target object item (which is in stock).

As described above, hierarchically clustering the purchasable target objects results in a hierarchical menu system, in which the target objects or clusters of target objects that appear on each menu can be labeled by names or icons and displayed in a two-dimensional or three-dimensional menu in which similar items are displayed physically near each other or on the same graphically represented "shelf." As described above, this grouping occurs both at the level of specific items (such as standard size Ivory soap or large Breck shampoo) and at the level of classes of items (such as soaps and shampoos). When the user selects a class of items (for instance, by clicking on it), then the more specific level of detail is displayed. It is neither necessary nor desirable to limit each item to appearing in one group; customers are more likely to find an object if it is in multiple categories. Non-purchasable objects such as artwork, advertisements, and free samples may also be added to a display of purchasable objects, if they are associated with (liked by) substantially the same users as are the purchasable objects in the display.

Network Context of the Browsing System

The files associated with target objects are typically distributed across a large number of different servers S1–So and clients C1–Cn. Each file has been entered into the data storage medium at some server or client in any one of a number of ways, including, but not limited to: scanning, keyboard input, e-mail, FTP transmission. automatic synthesis from another file under the control of another computer program. While a system to enable users to efficiently locate target objects may store its hierarchical cluster tree on a single centralized machine, greater efficiency can be achieved if the storage of the hierarchical cluster tree is distributed across many machines in the network. Each cluster C, including single-member clusters (target objects), is digitally represented by a file F, which is multicast to a topical multicast tree MT(C1); here cluster C1 is either cluster C itself or some supercluster of cluster C. In this way, file F is stored at multiple servers, for redundancy. The file F that represents cluster C contains at least the following data:

1. The cluster profile for cluster C, or data sufficient to reconstruct this cluster profile. 2. The number of target objects contained in cluster C. 3. A human-readable label for cluster C, as described in section "Labeling Clusters" above. 4. If the cluster is divided into subclusters, a list of pointers to files representing the subclusters. Each pointer is an ordered pair containing naming, first, a file, and second, a multicast tree or a specific server where that file is stored. 5. If the cluster consists of a single target object, a pointer to the file corresponding to that target object.

The process by which a client machine can retrieve the file F from the multicast tree MT(C 1) is described above in section "Retrieving Files from a Multicast Tree." Once it has retrieved file F, the client can perform further tasks pertaining to this cluster, such as displaying a labeled menu of subclusters, from which the user may select subclusters for the client to retrieve next.

The advantage of this distributed implementation is threefold. First, the system can be scaled to larger cluster sizes and numbers of target objects, since much more searching and data retrieval can be carried out concurrently. Second, the system is fault-tolerant in that partial matching can be achieved even if portions of the system are temporarily unavailable. It is important to note here the robustness due to redundancy inherent in our design—data is replicated at tree sites so that even if a server is down, the data can be located elsewhere.

The distributed hierarchical cluster tree can be created in a distributed fashion, that is, with the participation of many processors. Indeed, in most applications it should be recreated from time to time, because as users interact with target objects, the associative attributes in the target profiles of the target objects change to reflect these interactions; the system's similarity measurements can therefore take these interactions into account when judging similarity, which allows a more perspicuous cluster tree to be built The key technique is the following procedure for merging n disjoint cluster trees, represented respectively by files F1 . . . Fn in distributed fashion as described above, into a combined cluster tree that contains all the target objects from all these trees. The files F1 . . . Fn are described above, except that the cluster labels are not included in the representation. The following steps are executed by a server S1, in response to a request message from another server S0, which request message includes pointers to the files F1 . . . Fn. 1. Retrieve files F1 . . . Fn. 2. Let L and M be empty lists. 3. For each file Fi from among F1 . . . Fn: 4. If file Fi contains pointers

to subcluster files, add these pointers to list L. 5. If file Fi represents a single target object, add a pointer to file Fi to list L. 6. For each pointer X on list L, retrieve the file that pointer P points to and extract the cluster profile P(X) that this file stores. 7. Apply a clustering algorithm to group the pointers X on list L according to the distances between their respective cluster profiles P(X). 8. For each (nonempty) resulting group C of pointers: 9. If C contains only one pointer, add this pointer to list M; 10. otherwise, if C contains exactly the same subclusters pointers as does one of the files Fi from among F1 . . . Fn, then add a pointer to file Fi to list M; 11. otherwise: 12. Select an arbitrary server S2 on the network, for example by randomly selecting one of the pointers in group C and choosing the server it points to. 13. Send a request message to server S2 that includes the subcluster pointers in group C and requests server S2 to merge the corresponding subcluster trees. 14. Receive a response from server S2, containing a pointer to a file G that represents the merged tree. Add this pointer to list M. 15. For each file Fi from among F1 . . . Fn: 16. If list M does not include a pointer to file Fi, send a message to the server or servers storing Fi instructing them to delete file Fi. 17. Create and store a file F that represents a new cluster, whose subclusters pointers are exactly the subcluster pointers on list M. 18. Send a reply message to server S0, which reply message contains a pointer to file F and indicates that file F represents the merged cluster tree.

With the help of the above procedure, and the multicast tree MT full that includes all proxy servers in the network, the distributed hierarchical cluster tree for a particular domain of target objects is constructed by merging many local hierarchical cluster trees, as follows. 1. One server S (preferably one with good connectivity) is elected from the tree. 2. Server S sends itself a global request message that causes each proxy server in MT_{full} (that is., each proxy server in the network) to ask its clients for files for the cluster tree. 3. The clients of each proxy server transmit to the proxy server any files that they maintain, which files represent target objects from the appropriate domain that should be added to the cluster tree. 4. Server S forms a request R1 that, upon receipt, will cause the recipient server S1 to take the following actions: (a) Build a hierarchical cluster tree of all the files stored on server S1 that are maintained by users in the user base of S1. These files correspond to target objects from the appropriate domain. This cluster tree is typically stored entirely on S1, but may in principle be stored in a distributed fashion. (b) Wait until all servers to which the server S1 has propagated request R have sent the recipient reply messages containing pointers to cluster trees. (c) Merge together the cluster tree created in step 5(*a*) and the cluster trees supplied in step 5(*b*), by sending any server (such as S1 itself) a message requesting such a merge, as described above. (d) Upon receiving a reply to the message sent in (c), which reply includes a pointer to a file representing the merged cluster tree, forward this reply to the sender of request R1, unless this is S1 itself 5. Server S sends itself a global request message that causes all servers in MT_{full} to act on embedded request R1. 6. Server S receives a reply to the message it sent in 5(*c*). This reply includes a pointer to a file F that represents the completed hierarchical cluster tree. Server S multicasts file F to all proxy servers in MT_{full}. Once the hierarchical cluster tree has been created as above, server S can send additional messages through the cluster tree, to arrange that multicast trees MT(C) are created for sufficiently large clusters C, and that each file F is multicast to the tree MT(C), where C is the smallest cluster containing file F.

VIRTUAL COMMUNITIES AND THE VIRTUAL ORGANIZATION

Matching users for Virtual Communities on the Internet

Computer users frequently join other users for discussions on computer bulletin boards, newsgroups, mailing lists, and real-time chat sessions over the computer network, which may be typed (as with Internet Relay Chat (IRC)), spoken (as with Internet phone), or videoconferenced. These forums are herein termed "virtual communities." In current practice, each virtual community has a specified topic, and users discover communities of interest by word of mouth or by examining a long list of communities (typically hundreds or thousands). The users then must decide for themselves which of thousands of messages they find interesting from among those posted to the selected virtual communities, that is, made publicly available to members of those communities. If they desire, they may also write additional messages and post them to the virtual communities of their choice. The existence of thousands of Internet bulletin boards (also termed newsgroups) and countless more Internet mailing lists and private bulletin board services (BBS's) demonstrates the very strong interest among members of the electronic community in forums for the discussion of ideas about almost any subject imaginable. Presently, virtual community creation proceeds in a haphazard form, usually instigated by a single individual who decides that a topic is worthy of discussion. There are protocols on the Internet for voting to determine whether a newsgroup should be created, but there is a large hierarchy of newsgroups (which begin with the prefix "alt.") that do not follow this protocol.

The system for customized electronic identification of desirable objects described herein can of course function as a browser for bulletin boards, where target objects are taken to be bulletin boards, or subtopics of bulletin boards, and each target profile is the cluster profile for a cluster of documents posted on some bulletin board. Thus, a user can locate bulletin boards of interest by all the navigational techniques described above, including browsing and querying. However, this method only serves to locate existing virtual communities. Because people have varied and varying complex interests, it is desirable to automatically locate groups of people with common interests in order to form virtual communities. The Virtual Community Service (VCS) described below is a network-based agent that seeks out users of a network with common interests, dynamically creates bulletin boards or electronic mailing lists for those users, and introduces them to each other electronically via e-mail. It is useful to note that once virtual communities have been created by VCS, the other browsing and filtering technologies described above can subsequently be used to help a user locate particular virtual communities (whether pre-existing or automatically generated by VCS); similarly, since the messages sent to a given virtual community may vary in interest and urgency for a user who has joined that community, these browsing and filtering technologies (such as the e-mail filter) can also be used to alert the user to urgent messages and to screen out uninteresting ones.

The functions of the Virtual Community Service are general functions that could be implemented on any network ranging from an office network in a small company to the World Wide Web or the Internet. The four main steps in the procedure are: 1. Scan postings to existing virtual communities. 2. Identify groups of users with common interests. 3. Match users with virtual communities, creating new virtual communities when necessary. 4. Continue to enroll additional users in the existing virtual communities.

More generally, users may post messages to virtual communities pseudonymously, even employing different pseud-

onyms for different virtual communities. (Posts not employing a pseudonymous mix path may, as usual, be considered to be posts employing a non-secure pseudonym, namely the user's true network address.) Therefore, the above steps may be expressed more generally as follows: 1. Scan pseudonymous postings to existing virtual communities. 2. Identify groups of pseudonyms whose associated users have common interests. 3. Match pseudonymous users with virtual communities, creating new virtual communities when necessary. 4. Continue to enroll additional pseudonymous users in the existing virtual communities. Each of these steps can be carried out as described below.

Virtual Organization

E-mail Groupware on the Intranet (Intranet applications)

Another application of Virtual Communities is the application to virtual organizations. Organizations may use the above described techniques in accordance with their unique circumstances of intranet enabled communications involving telephony, voice and video conferencing, voice mail groupware and e-mail. By enabling users to better communicate, route messages by matching users together with each other or filtering e-mail or voice message, the following viable applications apply to the techniques of the previously described technologies including matching users in virtual communities on the Internet and those described in the previous sections.

E-mail Filter

In addition to the news clipping service described above, the system for customized electronic identification of desirable objects functions in an e-mail environment in a similar but slightly different manner. The news clipping service selects and retrieves news information that would not otherwise reach its subscribers. But at the same time, large numbers of e-mail messages do reach users, having been generated and sent by humans or automatic programs. These users need an e-mail filter, which automatically processes the messages received. The necessary processing includes a determination of the action to be taken with each message, including, but not limited to: filing the message, notifying the user of receipt of a high priority message, automatically responding to a message. The e-mail filter system must not require too great an investment on the part of the user to learn and use, and the user must have confidence in the appropriateness of the actions automatically taken by the system. The same filter may be applied to voice mail messages or facsimile messages that have been converted into electronically stored text, whether automatically or at the user's request, via the use of w ell-known techniques for speech recognition or optical character recognition.

The filtering problem can be defined as follows: a message processing function MPF(*) maps from a received message (document) to one or more of a set of actions. The actions, which may be quite specific, may be either predefined or customized by the use r. Each action A has an appropriateness function F_A (*,*) such that F_A (U,D) returns a real number, representing the appropriateness of selecting action A on behalf of user U when user U is in receipt of message D. For example, if D comes from a credible source and is marked urgent, then discarding the message has a high cost to the user and has low appropriateness, so that $F_{discard}$ (U,D) is small, whereas alerting the user of receipt of the message is highly appropriate, so that F_{alert} (U,D) is large. Given the determined appropriateness function, the function MPF(D) is used to automatically select the appropriate action or actions. As an example, the following set of actions might be useful:

1. Urgently notify user of receipt of message and/or insert message higher in the queue indicating its priority.
2. Insert message into queue for user to read later
3. Insert message into queue for user to read later, and suggest that user reply
4. Insert message into queue for user to read later, and suggest that user forward it to individual R where individual R's profile indicates that the message is relevant to himn/her or suggest that the message be sent as a voice mail using text to speech or as a fax or e-mail. The message may also be in the form of voice mail or voice e-mail.
5. Summarize message and insert summary into queue
6. Forward message to user's secretary
7. File message in directory X
8. File message in directory Y
9. Delete message (i.e., ignore message and do not save) and/or
10. Notify sender that further messages on this subject are unwanted
11. Provide a form auto request response that the sender of the e-mail (or voice mail) message will be ignored (and that it will be deleted).
12. Send a form auto response to the sender of an e-mail message that the user is out of town where the identity (or user profile) determines the selection of the response message.
13. Send a form auto response message to an individual to which the user does not want to directly reply to.
14. Similarly provide an auto response voice mail message that is specific (or most relevant) to the identity of the caller.
15. Suggest to the user to authorize a form auto request for deletion from a mailing list. Provide an automatic call screening function.
16. Provide an automatic call screening function wherein depending on the caller's identity to determine whether to allow the call to pass through to the secretary or user or to prompt the user to indicate the nature/purpose of his/her call using a speech to text conversion module to automatically select the most appropriate auto response message, whether to forward the call to the user's secretary, forward the call directly to the user, or automatically page the user, or request that the user not call back where these determinations are made based upon the identity of the caller and/or the stated objectives of the call or automatically forward the call to another user whose profile is more relevant. In this scenario if the user so desires if the call is forwarded directly to the user or if the user is paged while the caller is holding or if upon the system's determination it is forwarded to the user's voice mail, the user may identify the caller and/or listen to his/her stated objective of the call or automatically inform the caller based upon his/her identity and/or stated calling objective not to call back (where the voice mail option is not provided).
17. Notify user periodically that message "x" requests and warrants a reply due to its urgency and remind users periodically.
18. Automatically recommend to the user a mailing list of the most appropriate prospective recipients of a given outbound e-mail message. This list is determined by both the user's previous e-mail activities regarding those prospective recipients and their user profiles as well.

19. Accordingly suggest to the user a mailing list or automatically forward incoming e-mail messages which have been received wherein the user is not the most appropriate recipient for that message (if appropriate the forwarding party may also view the profile of the recommended recipient(s) prior to approving the recommendation). This system may also be used as an e-mail router for incoming e-mail or voice mail coming into an organization which occurs automatically or upon a human's approval.

The above appropriateness functions may of course instead first be manually entered as if then rules which are techniques well known in the art. Additionally, the automatically generated version of these rules (herein suggested) may be instead automatically written in which case the user may approve or rewrite a recommended appropriateness function (e.g., it may indicate that if the value of a specific word in the message exceeds value X perform appropriateness function Y).

Additional applications of the present methods are conceivable. For example in the case of sending, forwarding (or reforwarding) message to users based upon appropriateness functions relating to the profiles of the message and prospective recipients, it is possible to use this technique to allow users to more efficiently submit queries for response by users within any intranet, an inter-organizational intranet (extranet) or the Internet. An example application of the scenario is as follows:

1. A newbie submits a query by web or e-mail.
2. The engine shows the user a few answers such that similar newbie, query, answer triples have been highly rated. (One kind of answer consists of nothing but the URL of a helpful site!)
3. If the user finds these answers unsatisfactory, the engine takes note of this feedback. Then it goes to plan B, and finds a few experts such that the newbie, query, expert, time-of-day tuples have been highly rated.
4. The system offers all of these experts the question by e-mail.
5. First expert to indicate interest in the offer (by replying "yes") gets a go-ahead from the system.
6. Expert replies by sending an answer from the system. S/he may reply if further dialogue is needed—a conversation can continue in this way indefinitely. Of course, it all goes through the system, so it's all pseudonymous and logged. (Sometimes the correspondence may go off-topic. There should be a mechanism for dealing with this, so that rambling (or personal) discussion won't appear in it's entirety as part of that database. E.g., If I want to go off-topic with my next message itself The system then forwards the message as usual, but with my real return address as the Reply-to field. Further correspondence (if the other correspondent chooses to reply) then occurs with real names and outside the system.)
8. The newbie rates the quality of the dialogue, as a precondition for being allowed to ask more questions. (the expert is allowed to rate it too, so that the system knows which questions the expert LIKES to answer, not just which ones s/he WILL answer.)
9. If the dialogue never took place, because some expert replied in step 5 but didn't continue to step 6 within a reasonable time, the system sends a go-ahead to the next most appropriate of the experts who indicated interest in step 5. It also does this if the newbie got an answer but said (in step 8) that it was unhelpful. (In the latter case, the system might allow the newbie to edit the query first. The edited query would be included in the go-ahead to the next expert.)
10. If in step 5 or step 9 none of the (remaining) experts have indicated interest, within a reasonable time after the question was originally posed, then the system slowly offers the question to more experts (as in Step 4), up to a reasonable limit, until it does get a bite.
11. Any expert who received a request but ignored it gets a relevance feedback value of 0 for that query. Any expert who gave a go-ahead, but didn't get to answer, For choosing an expert, some interesting attributes of an expert are usual time to respond, length of response, count of technical term in response—since different users may have different sensitivities to these factors. Also the text of queries/list of queries they've answered, what clusters of newbies has rated them highly, etc. Finally, the set of terms in their explicit declarations of interest, and in their responses: this helps cluster them both with queries and with other experts. If we had a billing mechanism (which would probably require collaboration with AFL or someone, since its currently hard to collect from a user who only spends $1/month on queries), here would be a rough pricing model: When a question lands on your desk, it comes accompanied by an offer of payment. So the system looks for an expert, price pair such that newbie, query, expert, price, time-of-day id highly rated, meaning:
    - this expert is likely to answer this question for this price at this time
    - this newbie will be satisfied with the tradeoff between answer and price paid

This ought to work fine, in terms of getting offered prices to fluctuate correctly. It does mean that it's hard to lower your rates (in a particular area) once the system has decided you're expensive and stopped sending you queries, but there are ways around this. (e.g. you could always actively notify the system of your new approximate rates, either out of the blue or when responding to a request. In addition, the system might e-mail inactive experts every so often, asking if they want to lower their rates, declare additional interests, be dropped from the rolls, etc.). There is also a free-of-charge model, which is presumably the best way to start. It might involve some or all of these elements:

- Get nice idealists to participate as experts, by advertising on Usenet (and/or by actually seeding the database with Usenet postings from selected groups, so that people may be experts without knowing it). I think there are some people who would participate freely given that only a few people have to see each question, so they won't get many—it would reduce Usenet traffic, where everyone has to see all the questions—the answer would be permanently on file, and they could sign it (good for visibility!)
- if they ignore the questions they'll just go away.
- Attract advertising.
    - The benign kind of advertising: plugs in signs and on web sites
    - The sleazy kind: A query about word processors or WordPerfect is highly likely to draw an on-file "expert" response touting Microsoft Word
    - The semi-sleazy kind: the expert responses to the query are uncompromised (they're genuinely highly rated) but an Soft advert labeled such is attached (Apparently IBM bought the queries "Microsoft" and "gates" on Lycos!)

Use play money. By answering questions, you can build up credit that you can use to ask questions. However, if you go too deeply into debt, you have to fork over real money (or accept advertising). If you go well into profit, you can cash in. One could imagine eventually using this system as the seed of a VCS chat service, where queries consisted of topics advertisements. (We'd just have to allow new people to get added to existing conversations.) It's also a good way for consultants, brokers, mechanics, etc., to advertise their expertise (remember that answers can be paid for on-line, or a negotiation taken off-line). And for the same reason, I could all-too-easily imagine it replacing 1-900 phone sex numbers. (Hey-ratings, price and all!) This matching criteria includes interest attributes. Though this market model is useful for the above example it is readily applicable to any of the aforementioned applications (to retrieving information, human experts, employers and employees, buyers and sellers, and may be applied likewise to any product, commodity, share or interest that may be exchanged in an open market, e.g., stocks, commodities, insurance policies, products (bought and sold or bartered). Domains of application for the Internet-wide market system (such as legal counseling, medicine, engineering, psychological/sociological services, computer solutions) as well as more subjective domains such as architectural design, product design, document authoring, landscaping, decor (personalized fashion design) and cosmetics as well as informal solutions to problems of individuals based on their unique life and professional experiences, and encounters. Additionally, some experts may choose to use a filtering functionality on their system with preset parameters such as the price of a given task must meet a preset minimum to qualify.

Notice that actions 8 and 9 in the sample list above are designed to filter out messages that are undesirable to the user or that are received from undesirable sources, such as pesky salespersons, by deleting the unwanted message and/or sending a reply that indicates that messages of this type will not be read. The appropriateness functions must be tailored to describe the appropriateness of carrying out each action given the target profile for a particular document, and then a message processing function MPF can be found which is in some sense optimal with respect to the appropriateness function. One reasonable choice of MPF always picks the action with highest appropriateness, an d in cases where multiple actions are highly appropriate and are also compatible with each other, selects more than one action: for example, it may automatically reply to a message and also file the same message in directory X, so that the value of MPF(D) is the set\{reply, file in directory X\}. In cases where the appropriateness of even the most appropriate action falls below a user-specified threshold, as should happen for messages of an unfamiliar type, the system asks the user for confirmation of the action(s) selected by MPF. In addition, in cases where MPF selects one action over another action that is nearly as appropriate, the system also asks the user for confirmation: for example, mail should not be deleted if it is nearly as appropriate to let the user see it.

It is possible to write appropriateness functions manually, but the time necessary and lack of user expertise render this solution impractical. The automatic training of this system is preferable, using the automatic user profiling system described above. Each received document is viewed as a target object whose profile includes such attributes as the entire text of the document (represented as TF/IDF scores), document sender, date sent, document length, date of last document received from this sender, key words, list of other addressees, etc. It was disclosed above how to estimate an interest function on profiled target objects, using relevance feedback together with measured similarities among target objects and among users. In the con text of the e-mail filter, the task is to estimate several appropriateness functions $F_A(*,*)$, one per action. This is handled with exactly the same method as was used earlier to estimate the topical interest function f(*,*). Relevance feedback in this case is provided by the user's observed actions over time: whenever user U chooses action A on document D, either freely or by choosing or confirming an action recommended by the system, this is taken to mean that the appropriateness of action A on document D is high, particularly if the user takes this action A immediately after seeing document D. A presumption of no appropriateness (corresponding to the earlier presumption of no interest) is used so that action A is considered inappropriate on a document unless the user or similar users have taken action A on this document or similar documents. In particular, if no similar document has been seen, no action is considered especially appropriate, and the e-mail filter asks the user to specify the appropriate action or confirm that the action chosen by the e-mail filter is the appropriate one.

Thus, the e-mail filter learns to take particular actions on e-mail messages that 3-have certain attributes or combinations of attributes. For example, messages from John Doe that originate in the (212) area code may prompt the system to forward a copy by fax transmission to a given fax number, or to file the message in directory X on the user's client processor. A variation allows active requests of this form from the user, such as a request that any message from John Doe be forwarded to a desired fax number until further notice. This active user input requires the use of a natural language or form-based interface for which specific commands are associated with particular attributes and combinations of attributes.

Scanning

Using the technology described above, Virtual Community Service constantly scans all the messages posted to all the newsgroups and electronic mailing lists on a given network, and constructs a target profile for each message found. The network can be the Internet, or a set of bulletin boards maintained by America Online, Prodigy, or CompuServe, or a smaller set of bulletin boards that might be local to a single organization, for example a large company, a law firm, or a university. The scanning activity need not be confined to bulletin boards and mailing lists that were created by Virtual Community Service, but may also be used to scan the activity of communities that predate Virtual Community Service or are otherwise created by means outside the Virtual Community Service system, provided that these communities are public or otherwise grant their permission.

The target profile of each message includes textual attributes specifying the title and body text of the message. In the case of a spoken rather than written message, the latter attribute may b e computed from the acoustic speech data by using a speech recognition system. The target profile also includes an associative attribute listing the author(s) and designated recipient(s) of the message, where the recipients may be individuals and/or entire virtual communities; if this attribute is highly weighted, then the system tends to regard messages among the same set of people as being similar or related, even if the topical similarity of the messages is not clear from their content, as may happen when some of the messages are very short. Other important attributes include

the fraction of the message that consists of quoted material from previous messages, as well as attributes that are generally useful in characterizing documents, such as the message's date, length, and reading level.

Virtual Community Identification

Next, Virtual Community Service attempts to identify groups of pseudonymous users with common interests. These groups, herein termed "pre-communities," are represented as sets of pseudonyms. Whenever Virtual Community Service identifies a pre-community, it will subsequently attempt to put the users in said pre-community in contact with each other, as described below. Each pre-community is said to be "determined" by a cluster of messages, pseudonymous users, search profiles, or target objects.

In the usual method for determining pre-communities, Virtual Community Service clusters the messages that were scanned and profiled in the above step, based on the similarity of those messages' computed target profiles, thus automatically finding threads of discussion that show common interests among the users. Naturally, discussions in a single virtual community tend to show common interests; however, this method uses all the texts from every available virtual community, including bulletin boards and electronic mailing lists. Indeed, a user who wishes to initiate or join a discussion on some topic may send a "feeler message" on that topic to a special mailing list designated for feeler mess ages; as a consequence of the scanning procedure described above, the feeler message is automatically grouped with any similarly profiled messages that have been sent to this special mailing list, to topical mailing lists, or to topical bulletin boards. The clustering step employs "soft clustering," in which a message may belong to multiple clusters and hence to multiple virtual communities. Each cluster of messages that is found by Virtual Community Service and that is of sufficient size (for example, 10–20 different messages) determines a pre-community whose members are the pseudonymous authors and recipients of the messages in the cluster. More precisely, the pre-community consists of the various pseudonyms under which the messages in the cluster were sent and received.

Alternative methods for determining a pre-community, which do not require the scanning step above, include the following: 1. Pre-communities can be generated by grouping together users who have similar interests of any sort, not merely Individuals who have already written or received messages about similar topics. If the user profile associated with each pseudonym indicates the user's interests, for example through an associative attribute that indicates the documents or Web sites a user likes, then pseudonyms can be clustered based on the similarity of their associated user profiles, and each of the resulting clusters of pseudonyms determines a pre-community comprising the pseudonyms in the cluster. 2. If each pseudonym has an associated search profile set formed through participation in the news clipping service described above, then all search profiles of all pseudonymous users can be clustered based on their similarity, and each cluster of search profiles determines a pre-community whose members are the pseudonyms from whose search profile sets the search profiles in the cluster are drawn. Such groups of people have been reading about the same topic (or, more generally, accessing similar target objects) and so presumably share an interest. 3. If users participate in a news clipping service or any other filtering or browsing system for target objects, then an individual user can pseudonymously request the formation of a virtual community to discuss a particular cluster of one or more target objects known to that system. This cluster of target objects determines a pre-community consisting of the pseudonyms of users determined to be most interested in that cluster (for example, users who have search profiles similar to the cluster pro file), together with the pseudonym of the user who requested formation of the virtual community.

Matching Users with Communities

Once Virtual Community Service identifies a cluster C of messages, users, search profiles, or target objects that determines a pre-community M, it attempts to arrange for the members of this pre-community to have the chance to participate in a common virtual community V. In many cases, an existing virtual community V may suit the needs of the pre-community M. Virtual Community Service first attempts to find such an existing community V. In the case where cluster C is a cluster of messages, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the mean profile of the set of messages recently posted to virtual community V; in the case where cluster C is a cluster of users, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the mean user profile of the active members of virtual community V; in the case where the cluster C is a cluster of search profiles, V may be chosen to be any existing virtual community such that the cluster profile of cluster C is within a threshold distance of the cluster profile of the largest cluster resulting from clustering all the search profiles of active members of virtual community V; and in the case where the cluster C is a cluster of one or more target objects chosen from a separate browsing or filtering system, V may be chosen to be any existing virtual community initiated in the same way from a cluster whose cluster profile in that other system is within a threshold distance of the cluster profile of cluster C. The threshold distance used in each case is optionally dependent on the cluster variance or cluster diameter of the profile sets whose means are being compared.

If no existing virtual community V meets these conditions and is also willing to accept all the users in pre-community M as new members, then Virtual Community Service attempts to create a new virtual community V. Regardless of whether virtual community V is an existing community or a newly created community, Virtual Community Service sends an e-mail message to each pseudonym P in pre-community M whose associated user U does not already belong to virtual community V (under pseudonym P) and has not previously turned down a request to join virtual community V. The e-mail message informs user U of the existence of virtual community V, and provides instructions which user U may follow in order to join virtual community V if desired; these instructions vary depending on whether virtual community V is an existing community or a new community. The message includes a credential, granted to pseudonym P, which credential must be presented by user U upon joining the virtual community V, as proof that user U was actually invited to join. If user U wishes to join virtual community V under a different pseudonym Q, user U may first transfer the credential from pseudonym P to pseudonym Q, as described above. The e-mail message further provides an indication of the common interests of the community, for example by including a list of titles of messages recently sent to the community, or a charter or introductory message provided by the community (if available), or a label generated by the methods described above that identifies the content of the cluster of messages, user profiles, search profiles, or target objects that was used to identify the pre-community M.

If Virtual Community Service must create a new community V, several methods are available for enabling the members of the new community to communicate with each other. If the pre-community M is large, for example containing more than 50 users, then Virtual Community Service typically establishes either a multicast tree, as described below, or a widely-distributed bulletin board, assigning a name to the new bulletin board. If the pre-community M has fewer members, for example 2–50, Virtual Community Service typically establishes either a multicast tree, as described below, or an e-mail mailing list. If the new virtual community V was determined by a cluster of messages, then Virtual Community Service kicks off the discussion by distributing these messages to all members of virtual community V. In addition to bulletin boards and mailing lists, alternative for that can be created and in which virtual communities can gather include real-time typed or spoken conversations (or engagement or distributed multi-user applications including video games) over the computer network and physical meetings, any of which can be scheduled by a partly automated process wherein Virtual Community Service requests meeting time preferences from all members of the pre-community M and then notifies these individuals of an appropriate meeting time.

For multi user applications, users may be matched together who share a high level of interest in that application or the particular type of content therein as with educational software, entertainment applications or groupware (e.g., intra-organizational) where users may participate remotely in an application. Any of these multi-user applications may involve automatic calendaring (by a scheduling agent) for the purpose of arranging a virtual session between users who share a common interest in the nature or content of the application (e.g., a high speed action or suspense adventure video game) or for some applications (e.g., document editing groupware) users may sometimes require synchronous sessions or they may participate asynchronously. Conversely, users who are currently engaged in a multi user session may allow the VCS agent to notify or page remote users who may be interested in participating as in entertainment type applications or whose presence (or contribution) they feel is needed as with groupware used in an organizational or professional context (such as with on-line conferencing, whiteboarding, document editing, virtual corporate meetings, etc.). Matching together users in these applications assumes that within the current session, prospective participants share the same (or similar) application thus are profiled accordingly to the nature of the application, the list of current participants and if relevant secondarily to the content of the interacting user's dialogs (such as text or voice chat).

Specifically users are likely to have a common interest in the nature of an application which can be jointly (passively) interacted with or jointly viewed such as the content of the document being edited, the profile of a video being viewed or a site being visited by a group of users collaboratively navigating the WWW (or intra-organizational Web). A useful approach to advertising in a virtual chat room, conference or multi-user application is using the current temporal profile of the collaborative interaction as a target profile for which to target ads in real time and dynamically change the ad presentation as the topical relevance of the interaction changes, which is then viewed by all of the collaborative participants simultaneously. In a variation using similar techniques to those used in the above e-mail filter section, one appropriateness function which the system could write could be recommending to a user (such as an employer) whether or not a virtual meeting (or transcript thereof) should be made accessible to each employee in the organization based upon access privileges to particular types of content granted in the past and other aspects of his/her profile. This technique may be applied more generally as well to augment access control to information by employees in the organization in general.

In accordance with currently used methods, voice and fax numbers may change dynamically in accordance with the user's physical location. Specifically users should first be matched according to their common interest in a type of application which can be jointly interacted with or jointly viewed passively (via PC or TV). Then, secondly, users within such a common interest group may be further subdivided into sub-communities according to more specific common interests which they share (such as sub-communities) of real time correspondents simultaneously watching a popular program on television or according to content profile of the real time dialogues which the users are engaged in e.g., as they jointly navigate the World Wide Web, view a video program or television debate or engage in a video game. Conversely where the forum is smaller and/or the objectives are more objectively identified, sub-interest groups may be irrelevant, for example, on-line seminars, organizational meetings or board meetings in which relevant users whose presence or participation is requested may be automatically scheduled (by a scheduling agent) in advance or the user may be notified or paged if topical relevancy to the user's interest (or professional interest) profile is identified in real time by the VCS agent initially (or throughout the course of the meeting).

Continued Enrollment

Even after creation of a new virtual community, Virtual Community Service continues to scan other virtual communities for new messages whose target profiles are similar to the community's cluster profile (average message profile). Copies of any such messages are sent to the new virtual community, and the pseudonymous authors of these messages, as well as users who show high interest in reading such messages, are informed by Virtual Community Service (as for pre-community members, above) that they may want to join the community. Each such user can then decide whether or not to join the community. In the case of Internet Relay Chat (IRC), if the target profile of messages in a real time dialog are (or become) similar to that of a user, VCS may also send an urgent e-mail message to such user whereby the user may be automatically notified as soon as the dialog appears, if desired.

With these facilities, Virtual Community Service provides automatic creation of new virtual communities in any local or wide- area network, as well as maintenance of all virtual communities on the network, including those not created by Virtual Community Service. The core technology underlying Virtual Community Service is creating a search and clustering mechanism that can find articles that are "similar" in that the users share interests. This is precisely what was described above. One must be sure that Virtual Community Service does not bombard users with notices about communities in which they have no real interest. On a very small network a human could be "in the loop", scanning proposed virtual communities and perhaps even giving them names. But on larger networks Virtual Community Service has to run in fully automatic mode, since it is likely to find a large number of virtual communities.

Delivering Messages to a Virtual Community

Once a virtual community has been identified, it is straightforward for Virtual Community Service to establish

a mailing list so that any member of the virtual community may distribute e-mail to all other members. Another method of distribution is to use a conventional network bulletin board or newsgroup to distribute the messages to all servers in the network, where they can be accessed by any member of the virtual community. However, these simple methods do not take into account cost and performance advantages which accrue from optimizing the construction of a multicast tree to carry messages to the virtual community. Unlike a newsgroup, a multicast tree distributes messages to only a selected set of servers, and unlike an e-mail mailing list, it does so efficiently.

A separate multicast tree MT(V) is maintained for each virtual community V, by use of the following four procedures. 1. To construct or reconstruct this multicast tree, the core servers for virtual community V are taken to be those proxy servers that serve at least one pseudonymous member of virtual community V. Then the multicast tree MT(V) is established via steps 4–6 in the section "Multicast Tree Construction Procedure" above. 2. When a new user joins virtual community V, which is an existing virtual community, the user sends a message to the user's proxy server S. If user's proxy server S is not already a core server for V, then it is designated as a core server and is added to the multicast tree MT(V), as follows. If more than k servers have been added since the last time the multicast tree MT(V) was rebuilt, where k is a function of the number of core servers already in the tree, then the entire tree is simply rebuilt via steps 4–6 in the section "Multicast Tree Construction Procedure" above. Otherwise, server S retrieves its locally stored list of nearby core servers for V, and chooses a server S1. Server S sends a control message to S1, indicating that it would like to be added to the multicast tree MT(V). Upon receipt of this message, server S1 retrieves its locally stored subtree G1 of MT(V), and forms a new graph G from G1 by removing all degree-1 vertices other than S1 itself. Server S1 transmits graph G t o server S, which stores it as its locally stored subtree of MT(V). Finally, server S sends a message to itself and to all servers that are vertices of graph G, instructing these servers to modify their locally stored subtrees of MT(V) by adding S as a vertex and adding an edge between S1 and S. 3. When a user at a client q wishes to send a message F to virtual community V, client q embeds message F -in a request R instructing the recipient to store message F locally, for a limited time, for access by member s of virtual community V. Request R includes a credential proving that the user is a member of virtual community V or is otherwise entitled to post messages to virtual community V (for example is not "black marked" by that or other virtual community members). Client q then broadcasts request R to all core servers in the multicast tree MT(V), by means of a global request message transmitted to the user's proxy server as described above. The core servers satisfy request R, provided that they can verify the included credential. 4. In order to retrieve a particular message sent to virtual community V, a user U at client q initiates the steps described in section "Retrieving Files from a Multicast Tree," above. If user U does not want to retrieve a particular message, but rather wants to retrieve all new messages sent to virtual community V, then user U pseudonymously instructs its proxy server (which is a core server for V) to send it all messages that were multicast to MT(V) after a certain date. In either case, user U must provide a credential proving user U to be a member of virtual community V, or otherwise entitled to access messages on virtual community V.

APPENDED COLLABORATIVE COMPUTING APPLICATIONS

1. Automatic Retrieval and Assembly of Work Groups

A company often requires a team of skilled personnel (whose qualifications are specifically suited to the task at hand). For large corporations it is difficult to keep track of skill sets of its own internal employees. Conversely for small companies finding such skill outside is often essential. The present system may thus organize such groups accordingly. The purpose of the system is to emulate expert human organizers of work teams and make recommendations as to the most appropriately qualified team of available people for the given task at hand based upon the stated objectives and required tasks of a prospective project. Using the presently described technique using relevance feedback it is possible to match the profile of the project with that of the available pool of individuals. The organizer may wish to keep in mind a variety of considerations in selecting teams for example considering a variety of qualifications, psychographics and attributes pertaining to the user's profile as developed from his/her professional on-line activities and interactions. In view of the fact that some skill requirements exist in overlapping disciplines, that the more diversity (complementarity) of skills of its members, the greater the likelihood of covering the (important) skill requirements adequately (suggesting that the greater the complementarity of attributes characterizing the users base of qualifications and information content interaction the more synergistic the work process). Another consideration may be to find the fewest number of individuals as possible who collectively cover the apparent skill requirements. Still another consideration is to favor the reorganizing of groups which had previously proved themselves by arriving at a successful solution or product to a similar problem or task. Through out the work process certain sub-problems may require temporary consultation with appropriately qualified individuals who are more qualified than members of the present team. Each member of a virtual work group (whether intra-organizational or inter-organizational) maybe prescribed attributes by a superior such as credentials, observed skill sets from past experiences and psychographics. This method may also be used to observe patterns relating to what types of users are granted access to what type of informational content e.g., some members of a team may be made privy to some information which others are not in accordance with the methods suggested above. The system may present recommendations which restrict what types of data can be accessed by that user. Some restriction attributes may be explicit indicating documents containing which word attributes a user is forbidden to access. For others the restriction may be based upon explicit criteria for example including documents containing words which tend to co-occur (are metrically close) to those explicitly mentioned. Or relative attribute weighting values may be used as thresholds for determining automatically user document access and privileges. Another (appropriateness function) based criteria which may be used as well is the similarity measure between the document and user profiles. In this case it may be useful to automatically generate explicit rules which may present the user profile (with relative attribute weights) as well as that of the document to the authorized decision maker. Additionally, as suggested in the e-mail filter section (above) a fully trained system may additionally automatically present the rules (appropriateness functions) which it has written through passive training. Thus the user may again (in this case for automatically determining document access privileges) approve the rules presented or

modify them accordingly. Documents may also in some cases be retrievable in segments (which lack forbidden terms) if the authorized credential granting party so allows. In the case where documents or document segments and the corresponding individuals are ascribed manual restriction attributes, user restriction attributes may act as a restriction for either adding to or deleting from relevant documents (or segments) or prohibiting access altogether as suggested. These restrictions may be integrated with the document file such that authorization credentials may act as a decryption key should the document be transmitted or conveyed elsewhere (e.g., outside of the data base) enabling these access restrictions to apply to any users accessing that information anywhere. As suggested, it can be appreciated that the present technique can be usefully applied to the above applications of the virtual dialogues (live or indexed recorded) i.e., matching users with virtual meetings and the above e-mail and telephony router in both cases wherein users are granted attribute based privileges to access (or denial for accessing) certain dialogues in accordance with their content.

In one exemplary approach, a virtual work group is assembled for engineering a product, in another authoring or editing a document, in another arrive at corporate policy for a particular need or unresolved issue or for the purposes of creating virtual breakout sessions within an on-line conference (multi organizational) or corporate meeting. Many other examples are possible.

2. Virtual Meetings

Particularly within large organizations, it is advantageous to disseminate company (inside) news and information to those employees for whom the information is "valuable". Using the same basic profiling techniques (above). Virtual dialogues (either physical meetings or entirely virtual meetings, either e-mail or telephony based) may be automatically profiled on the fly and used for responsive indexing and notification of those users to whom the information is valuable (and to whom it is privy). As the content of such a dialogue may change with time, new users may be prompted to join while others may be prompted or alternatively (for confidentiality reasons) may be mandated to depart. Text summarization techniques may also be used to allow relevant users who missed the virtual meeting to have access to a synopsized version thereof. Document profiles of such meetings may also be organized into a hierarchical cluster tree using automatic cluster labeling or relevant terms within each cluster (Steve's reference hierarchical cluster menu trees from previous patent). This technique is useful for intuitive browsing of large archives of this information). Digital credentials may be prescribed to each employee by superiors which indicate for him/her the specific information contexts (by clusters) which are mandatory, which are recommended, which are neutral, and which are inappropriate for the employee to either access or (for the mandatory credential) require also mandatory (real-time) attendance. A scheduling agent maybe used to organize meeting times in advance by contacting and informing the most relevant users as to the stated objectives of the meeting. This is done by coordinating available time slots to optimize the availability of the most number of user highest relevance users to the dialogue (the user may also indicate among his/her available time the level of convenience as well). As above suggested, in virtual work groups a virtual meeting's objective may be to solve a particular problem, and develop a strategy, plan or proposal the stated objective of which may be used to index a virtual group whose complement and skills provides an optimal solution thereto.

3. Monitoring Dialogues

The above present methods may be used for retrieving documents by organizations to determine the relevance of internal correspondence (e-mail, fax, telephony and recorded physical dialogues) to the interests of the user as stated or exemplified. Thus all irrelevant correspondences are filtered out. Relevant ones may accordingly be clustered (labeled) and organized into a hierarchical cluster menu tree for industrial browsing as above described. For example, an employer may wish to "listen in" on certain types of correspondences with a particular client by a particular employee (via phone number and voice ID using Neural Net techniques) or about a particular topic. Again text summarization may aid the user in viewing large correspondences. In one approach fax, e-mail and telephone communications to and from each individual may also be monitored and advised similarly in order to enable the system to develop aggregate profiles for a given employee for both outgoing and incoming forms of each desired communication media which is used for purposes of routing.

A supervisor may designate particular clusters to be directly relevant, indirectly relevant or irrelevant to a given user's employment duties. For any employee a summary report of his/her work profile may be automatically e-mailed periodically and/or notification made if, for example, a certain irrelevant or indirectly relevant cluster exceeds a certain threshold or it may notify the superior if "unusual" patterns are detected or manually entered key word detection may be used in certain instances. In this regard, the attribute of time may be usefuil in determining whether or not irrelevant dialogues are occurring during scheduled work times. Length and frequency of the correspondences are additional useful attributes. Each cluster of interest may also be broken down to reveal the full profile of each associated correspondence. In an application variation the present technique may be used for purposes of monitoring activities and general behavior of children by parents or on-line scholastic (navigation) behavior by teachers. Phone companies may also apply this technique to better monitor communications channels for suspicious activities. In each of the above applications, an additional or alternative feature is the ability of the authoritative party to place restrictions on particular domains. These may be either explicitly mentioned attributes or examples or those which are metrically "similar" to the same. In one variation, a caller's identity (via incoming phone number and/or Neural Net based voice ID) is determinable.

4. Virtual Classroom

In one approach school activities (from either one or a large number of schools) may be accessible for participation remotely. Classroom lectures, continuing education seminars, conferences, tutorials for job training (or on-going job training requirements) may apply. The most exemplary application however is the virtual classroom. Students may use nearest neighbor indexing to either describe or present a particular topic or problems or a query. The system will recommend the most appropriate on-line lecture either live, if the student wishes to interact (e.g., recommending the next scheduled time) or the most appropriate pre-recorded lecture. For solutions to problems, a virtual tutor involving (either a live or pre-recorded single (closed) session or multi-student session may be presented similarly) or the student may receive a recommendation of the name of the most skilled or experienced faculty or student recommended tutor. In the classroom application the student may either present questions on-line to the lecturer (throughout the lecture or at pre-designated intervals) or the best ones may be selected by a moderator.

Additionally, if/when desired, sub-dialogues may occur between attendees in the absence of the others. This is also one application of joint user navigation as the presenter (lecturer or student presenting a question) presents questions, content or solutions or navigates through informational spaces in joint collaborative fashion for all attendees (or those designated by the presenter).

In one variation students who are most in need of a definable domain (attribute/cluster) indicated by their request or lack of proficiency as evidenced in quiz or test scores may be matched with offer students or tutors who are proficient in those areas. In one approach students may be matched for purposes of collaborative study sessions in which priority is given to those which possess the greater degree of complementarity within their respective domains of proficiency/deficiency. The present clustering model may further facilitate the predictors accuracy of the content domains in which a student is expect to be proficient. For example, in the pure clustering model, it is possible to make associations between which domains a student is LIKELY to be proficient in according to areas of previous proficiency (within the same class of different ones based upon historical data from previous students). It can be appreciated that the present system may readily be applied also to corporate or professional application including organizational training sessions, continuing education or conference seminars.

5. Virtual Communities Developed Around Product Genres, Categories, or Items.

The most "interested" users for a particular topic or target object (e.g., or limited to selected exemplary target objects) may be automatically matched for a virtual dialogue which is accessible directly from the target object of interest while browsing. This virtual dialogue includes standard bbs, IRC, Internet telephone and video telephony. Applications include store front products (and categories), musical albums, movies, stocks (or mutual funds). In one approach the criteria for creating a virtual group of watching people one on one is to find among "similar" users the greatest degree of complimentarity (difference) in their respective experiences. Thus optimizing the conditions for the users to share invaluable knowledge between one another business venture, a regional or national economy.

6. (Ancillary Inclusion) Hybrid TV/PC

In TV units which have integrated dual mode capabilities for TV and PC functionalities simultaneously (e.g., viewing TV programming while sending/receiving e-mail) the VCS agent may be used not only to point the users to the most appropriate TV programming for their interest at any given time (selectively refer to and/or transcribe Home Video Club patent) but it may also bring the participating views of a program to the attention of each other thus allowing viewers to exchange comments or share perspectives about the programming before, during or after the program. Within these user circles VCs may further narrow the criteria of interacting users by their specific viewing profiles.

7. Physical Meetings

In one exemplary approach VCS organized communities may meet in physical forums (e.g., where all the members are required to live in a physically close proximity as a prerequisite for matching) for example organizing meetings or according to general criteria (e.g., socializing and gathering in a restaurant/night club, concert or movie theater) or alternatively wherein a human or machine designated theme is the basis for the community for example a meeting around a political or a community related issue, an item of common interest within a large organization, a vacation destination (which all of the members are likely to wish to visit in the future and wherein a date could be scheduled using a scheduling agent for a group tour). Such a community could for example, be developed around such a travel destination as part of a travel agent's Web site as a marketing pitch for soliciting a trip.

SUMMARY

A method has been presented for automatically selecting articles of interest to a user. The method generates sets of search profiles for the users based on such attributes as the relative frequency of occurrence of words in the articles read by the users, and uses these search profiles to efficiently identify future articles of interest. The methods is characterized by passive monitoring (users do not need to explicitly rate the articles), multiple search profiles per user (reflecting interest in multiple topics) and use of elements of the search profiles which are automatically determined from the data (notably, the TF/IDF measure based on word frequencies and descriptions of purchasable items). A method has also been presented for automatically generating menus to allow users to locate and retrieve articles on topics of interest. This method clusters articles based on their similarity, as measured by the relative frequency of word occurrences. Clusters are labeled either with article titles or with key words extracted from the article. The method can be applied to large sets of articles distributed over many machines.

It has been further shown how to extend the above methods from articles to any class of target objects for which profiles can be generated, including news articles, reference or work articles, electronic mail product or service descriptions, people (based on the articles they read, demographic data, or the products they buy), and electronic bulletin boards (based on the articles posted to them). A particular consequence of being able to group people by their interests is that one can form virtual communities of people of common interest, who can then correspond with one another via electronic mail.

I claim:

1. A method for providing a user with access to selected ones of a plurality of target object bulletin boards that are accessible via an electronic data transmission media, where said users are connected via user terminals and data communication connections to a server system which provides access to said electronic data transmission media, said method comprising the steps of:

automatically generating target profiles for target object bulletin boards that are accessible by said electronic data transmission media, each of said target profiles being generated from the contents of an associated one of said target object bulletin boards;

automatically generating at least one user target profile interest summary for a user at a user terminal, each said user target profile interest summary being generated from ones of said target object bulletin boards accessed by said user; and

enabling access to said plurality of target object bulletin boards accessible by said electronic data transmission media by users via said target profile, comprising:

automatically creating virtual communities of users of said target object bulletin boards, comprising:

scanning bulletin board postings to existing target object bulletin boards,

identifying groups of user identifications whose associated users have common interests,

matching users with other like inclined users to create a new target object bulletin board.

2. The method for providing a user with access to selected ones of a plurality of target object bulletin boards of claim 1, wherein said step of automatically creating further comprises:

dynamically creating electronic mailing lists for said users matched by said step of matching.

3. The method for providing a user with access to selected ones of a plurality of target object bulletin boards of claim 2, wherein said step of automatically creating further comprises:

automatically transmitting a notification to said users matched by said step of matching to identify said new target object bulletin board to said ones of said associated users.

4. The method for providing a user with access to selected ones of a plurality of target object bulletin boards of claim 1, wherein said step of automatically creating further comprises:

continuing to enroll additional users in said new target object bulletin board.

5. A method for providing a user with access to selected ones of a plurality of target object bulletin boards that are accessible via an electronic data transmission media, where said users are connected via user terminals and data communication connections to a server system which provides access to said electronic data transmission media, said method comprising the steps of:

automatically generating target profiles for target object bulletin boards that are accessible by said electronic data transmission media, each of said target profiles being generated from the contents of an associated one of said target object bulletin boards comprising:

generating a target profile comprising the cluster profile for a cluster of documents posted on said new target object bulletin board;

automatically generating at least one user target profile interest summary for a user at a user terminal, each said user target profile interest summary being generated from ones of said target object bulletin boards accessed by said user; and

enabling access to said plurality of target object bulletin boards accessible by said electronic data transmission media by users via said target profile.

6. A method of operating a network-based agent to seek out users of a network with common interests, where said users are connected via user terminals and data communication connections to a server system which provides access to an electronic data transmission media, comprising the steps of:

dynamically creating bulletin boards for said users, comprising:

scanning bulletin board postings to existing bulletin boards,

identifying a group of users who have common interests,

matching users with other like inclined users in said identified group to create a proposed new bulletin board.

7. The method of operating a network-based agent of claim 6 wherein said step of scanning bulletin boards comprises:

automatically generating target profiles for bulletin boards that are accessible by said electronic data transmission media, each of said target profiles being generated from the contents of an associated one of said bulletin boards.

8. The method of operating a network-based agent of claim 7, wherein said step of automatically generating target profiles comprises:

generating a target profile comprising the cluster profile for a cluster of documents posted on said bulletin boards.

9. The method of operating a network-based agent of claim 6 wherein said step of identifying a group of users comprises:

automatically generating at least one user target profile interest summary for a user at a user terminal, each said user target profile interest summary being generated from ones of said bulletin boards accessed by said user.

10. The method of operating a network-based agent of claim 6, wherein said step of automatically creating further comprises:

dynamically creating electronic mailing lists for said users matched by said step of matching.

11. The method of operating a network-based agent of claim 10, wherein said step of automatically creating further comprises:

automatically transmitting a notification to said users matched by said step of matching to identify said proposed new bulletin board to said ones of said associated users.

12. The method of operating a network-based agent of claim 6, wherein said step of automatically creating further comprises:

continuing to enroll additional users in said proposed new bulletin board.

13. The method of operating a network-based agent of claim 6, wherein said step of matching comprises:

identifying an existing bulletin board whose mean profile of the set of messages recently posted therein is within a threshold distance of the cluster profile of said proposed new bulletin board.

14. The method of operating a network-based agent of claim 13, further comprising the step of:

automatically transmitting a notification to said users matched by said step of matching to identify said existing bulletin board to said ones of said associated users.

15. The method of operating a network-based agent of claim 14, wherein said step of automatically transmitting a notification comprises:

transmitting to said users matched by said step of matching an indication at least one of the data comprising an indication of common interest including: a list of titles of messages recently sent to the bulletin board, an introductory message provided by the bulletin board, a label that identifies the content of the cluster profile that was used to identify the existing bulletin board.

* * * * *

US006029182A

United States Patent [19]

Nehab et al.

[11] Patent Number: 6,029,182

[45] Date of Patent: Feb. 22, 2000

[54] **SYSTEM FOR GENERATING A CUSTOM FORMATTED HYPERTEXT DOCUMENT BY USING A PERSONAL PROFILE TO RETRIEVE HIERARCHICAL DOCUMENTS**

[75] Inventors: **Smadar Nehab**, Palo Alto; **Manjula G. Wickramaratne**, Fremont; **Paul L. Klark**, Mountain View, all of Calif.

[73] Assignee: **Canon Information Systems, Inc.**, Irvine, Calif.

[21] Appl. No.: **08/726,853**

[22] Filed: **Oct. 4, 1996**

[51] **Int. Cl.**[7] **G06F 17/30**

[52] **U.S. Cl.** **707/523**; 707/501

[58] **Field of Search** 707/523, 501; 345/349

[56] **References Cited**

U.S. PATENT DOCUMENTS

4,959,769	9/1990	Cooper et al.	707/200
4,965,763	10/1990	Zamora	704/1
5,181,162	1/1993	Smith et al.	707/530
5,267,155	11/1993	Buchanan et al.	707/540
5,327,554	7/1994	Palazzi, III et al.	348/13
5,347,632	9/1994	Filepp et al.	395/200.32
5,392,428	2/1995	Robins	707/3
5,408,655	4/1995	Oren et al.	707/501
5,423,043	6/1995	Fitzpatrick et al.	345/351
5,530,852	6/1996	Meske, Jr. et al.	395/200.36
5,649,186	7/1997	Ferguson	707/533
5,737,560	4/1998	Yohanan	345/349
5,754,939	5/1998	Hertz et al.	395/200.49
5,758,361	5/1998	Van Hoff	707/513
5,761,662	6/1998	Dasan	707/10
5,764,906	6/1998	Edelstein et al.	395/200.49
5,877,766	3/1999	Bates et al.	345/357
5,886,683	3/1999	Tognazzini et al.	345/146
5,890,152	3/1999	Rapaport et al.	707/6

OTHER PUBLICATIONS

Collins, Regina S., ed., "Journalist(TM): Your Personal Newspaper for Compuserve(R)", Cupertino, Ca.: PED Software Corporation, pp. 1–143, Jan. 1993.

Advertisement by the San Jose Mercury, "Newshound User Guide", no date available.

Adverstisement by Dow Jones and Company, "News Retrieval for Windows", no date available.

Beretta, Giordano, "W^3+Structure=Knowledge", Hewlett Packard Laboratories Technical Report HPL–96–99, Jun., 1996.

Online User, "Personal Journal—Daily News on Your Virtual Doorstep", pp. 50–54, Oct./Nov. 1995.

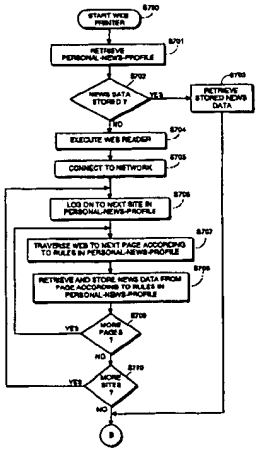
At www.e.g.bucknell.edu/boulter/crayon "Crayon—Create Your Own Newspaper", Jun. 26, 1995.

Primary Examiner—Mark R. Powell
Assistant Examiner—J. A. Rossi
Attorney, Agent, or Firm—Fitzpatrick, Cella, Harper & Scinto

[57] **ABSTRACT**

A World Wide Web site data retrieval system includes an input device for inputting data and commands to access the World Wide Web, and a memory for storing a Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information. The memory also stores process steps to connect to a Web site and to issue commands within the connected Web site, and a connection to the World Wide Web. The system includes a processor for launching the Web site data retrieval driver in response to a command to access the World Wide Web. The Web site retrieval driver, upon being launched, (1) launches the Web reader to connect to the World Wide Web via the connection, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, (5) stores the Web site data in a linear document, (6) repeats steps 1 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into a personalized document based on the format information.

21 Claims, 16 Drawing Sheets



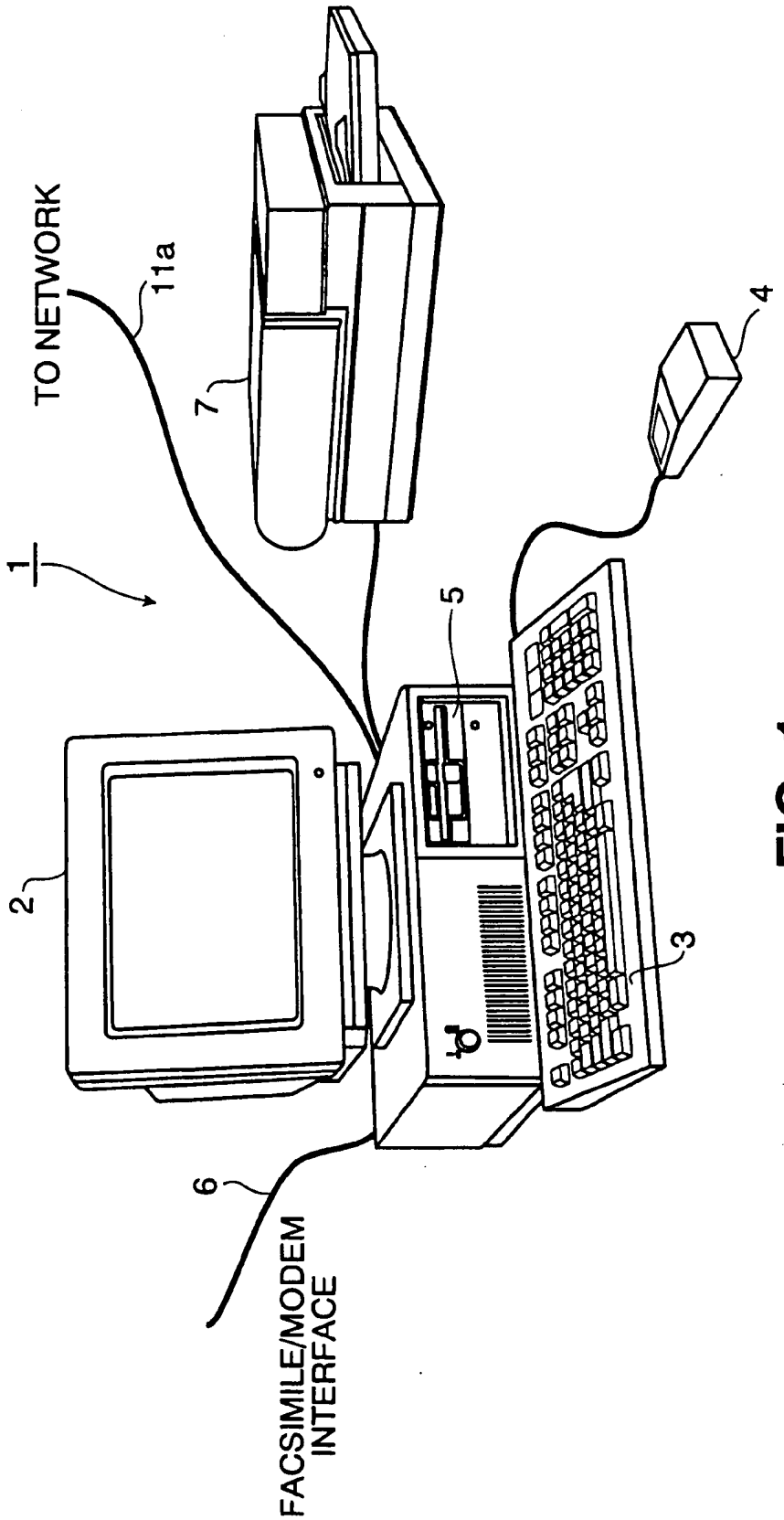


FIG. 1

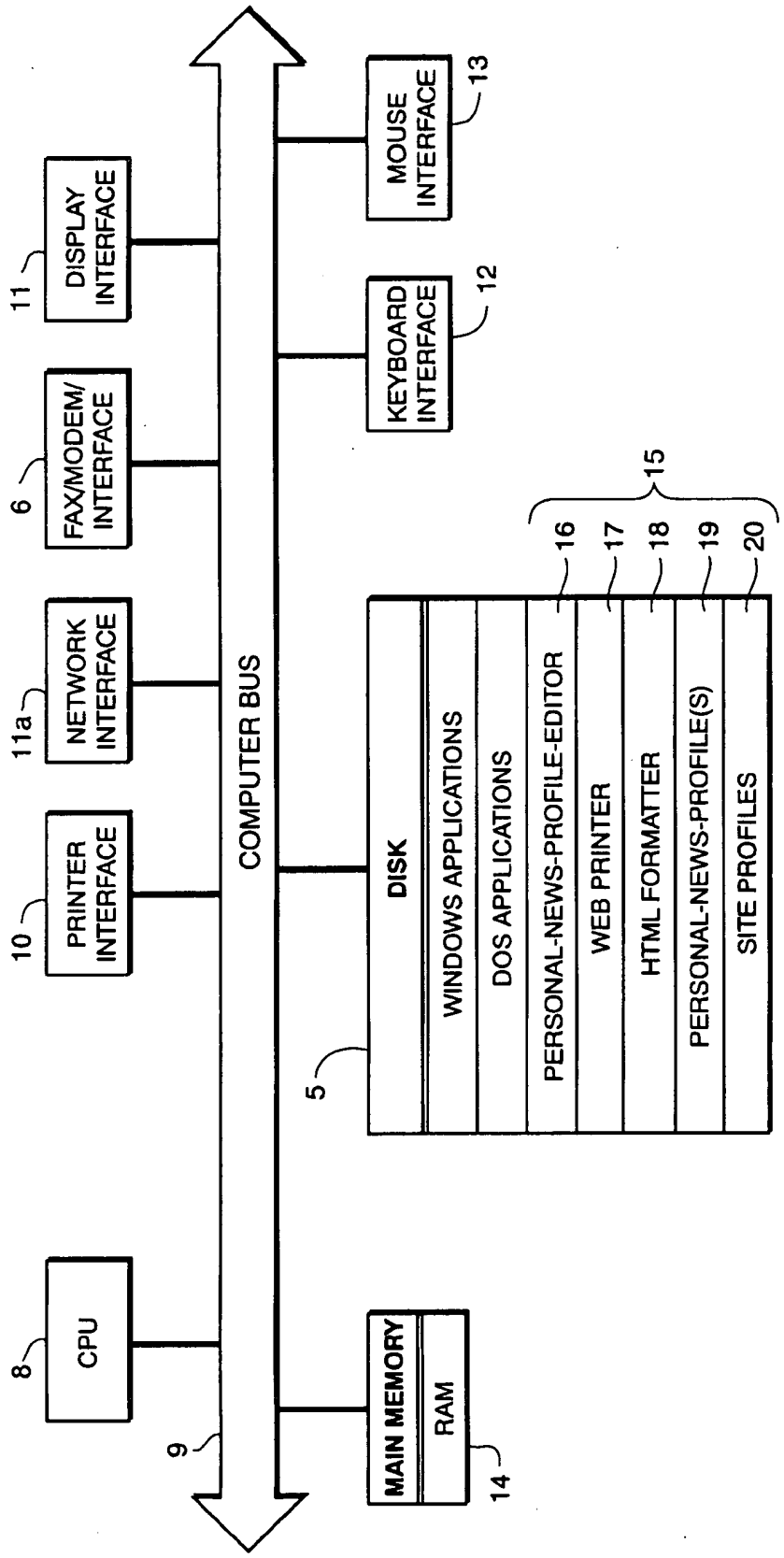

COMPUTER BUS
CPU
8
9
PRINTER INTERFACE
10
NETWORK INTERFACE
11a
FAX/MODEM/ INTERFACE
6
DISPLAY INTERFACE
11
MOUSE INTERFACE
13
KEYBOARD INTERFACE
12
DISK
5
WINDOWS APPLICATIONS
DOS APPLICATIONS
PERSONAL-NEWS-PROFILE-EDITOR
16
WEB PRINTER
17
HTML FORMATTER
18
PERSONAL-NEWS-PROFILE(S)
19
SITE PROFILES
20
15
MAIN MEMORY
RAM
14

FIG. 2

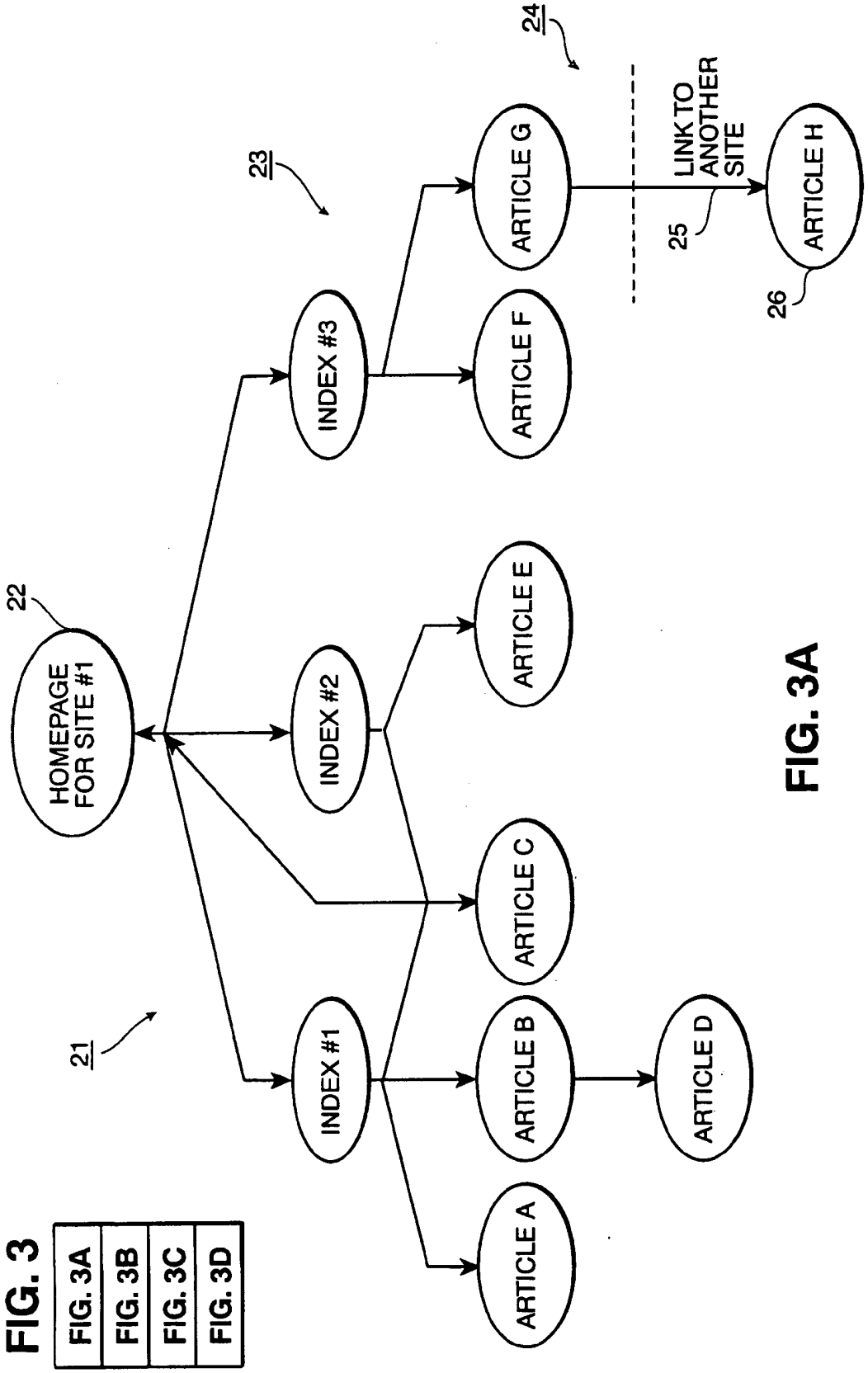


FIG. 3A

FIG. 3

FIG. 3A	FIG. 3B	FIG. 3C	FIG. 3D

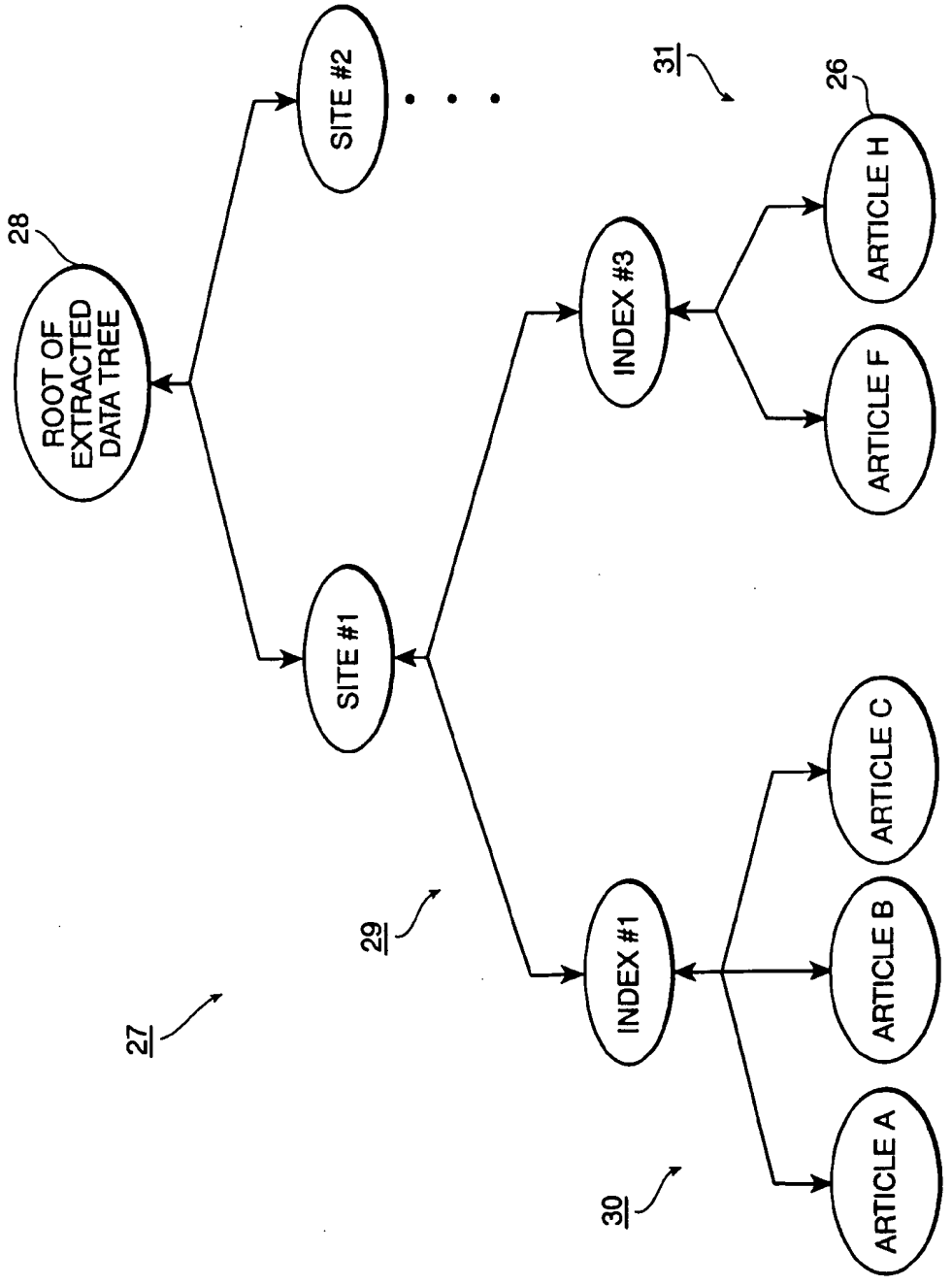

ROOT OF EXTRACTED DATA TREE
28
SITE #2
SITE #1
INDEX #3
INDEX #1
ARTICLE H
ARTICLE F
ARTICLE C
ARTICLE B
ARTICLE A
26
27
29
30
31

FIG. 3B

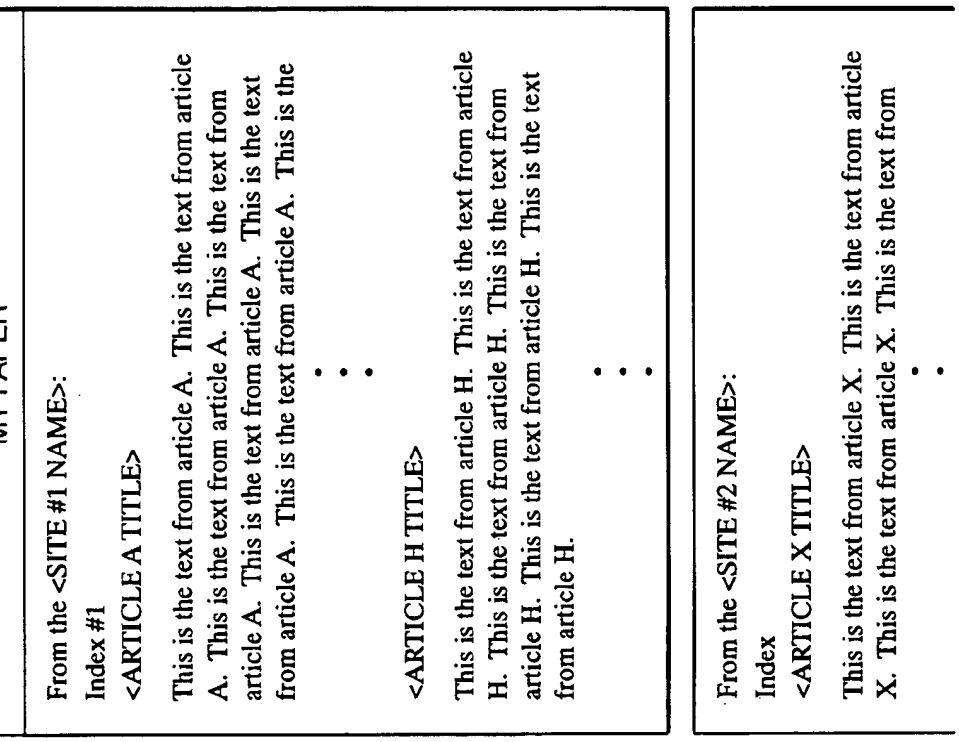

33
MY PAPER
From the <SITE #1 NAME>:
Index #1
<ARTICLE A TITLE>
This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the text from article A. This is the
<ARTICLE H TITLE>
This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H. This is the text from article H.
From the <SITE #2 NAME>:
Index
<ARTICLE X TITLE>
This is the text from article X. This is the text from article X. This is the text from article X. This is the text from

FIG. 3D

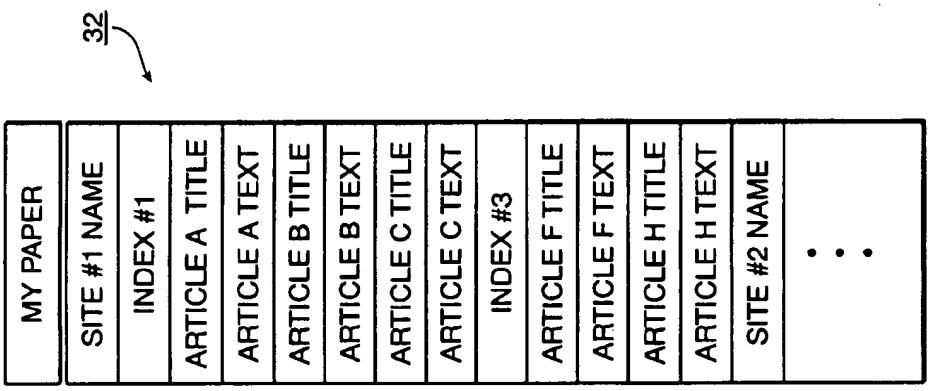

32
MY PAPER
SITE #1 NAME
INDEX #1
ARTICLE A TITLE
ARTICLE A TEXT
ARTICLE B TITLE
ARTICLE B TEXT
ARTICLE C TITLE
ARTICLE C TEXT
INDEX #3
ARTICLE F TITLE
ARTICLE F TEXT
ARTICLE H TITLE
ARTICLE H TEXT
SITE #2 NAME

FIG. 3C

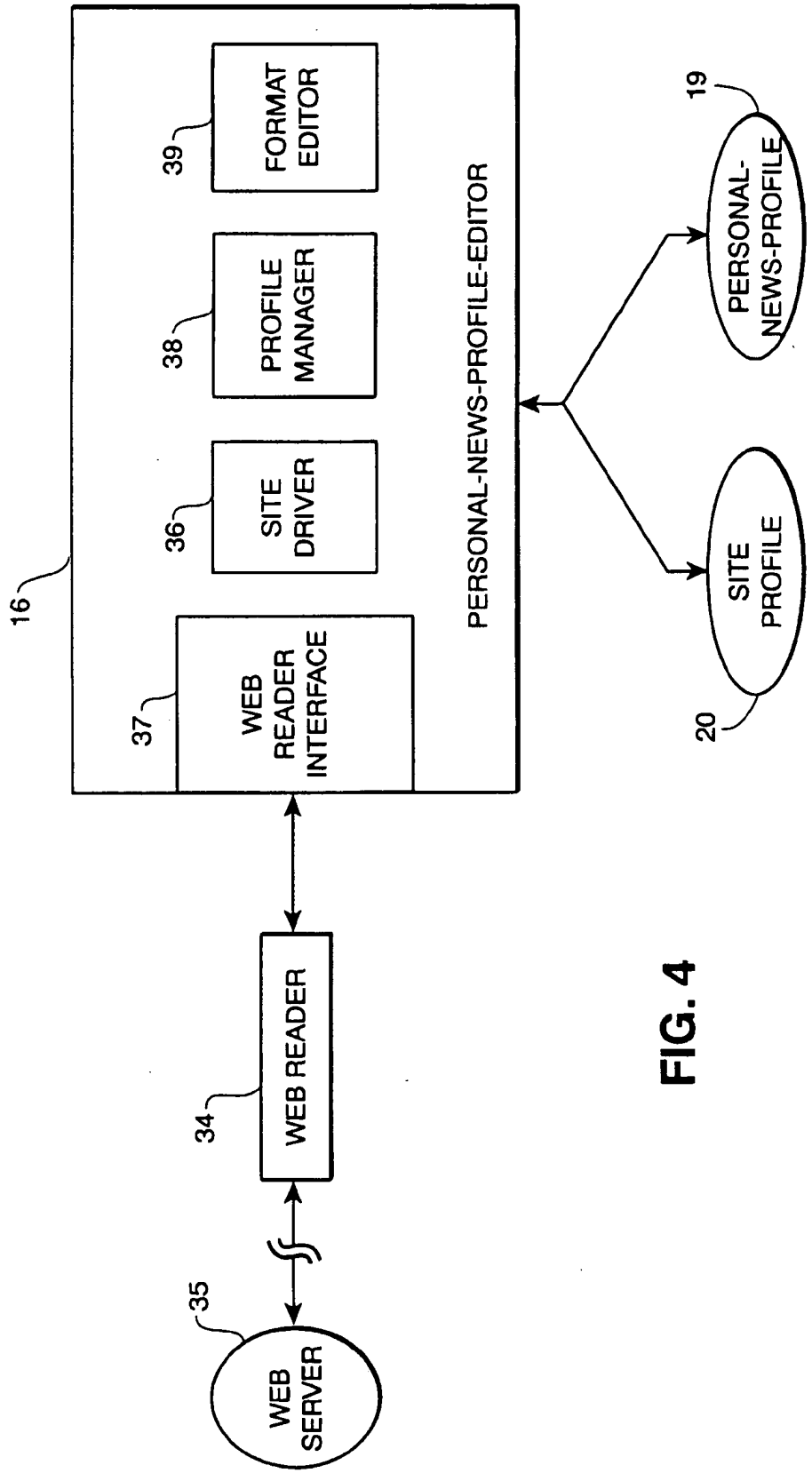

16
PERSONAL-NEWS-PROFILE-EDITOR
37
WEB READER INTERFACE
36
SITE DRIVER
38
PROFILE MANAGER
39
FORMAT EDITOR
19
PERSONAL-NEWS-PROFILE
20
SITE PROFILE
34
WEB READER
35
WEB SERVER

FIG. 4

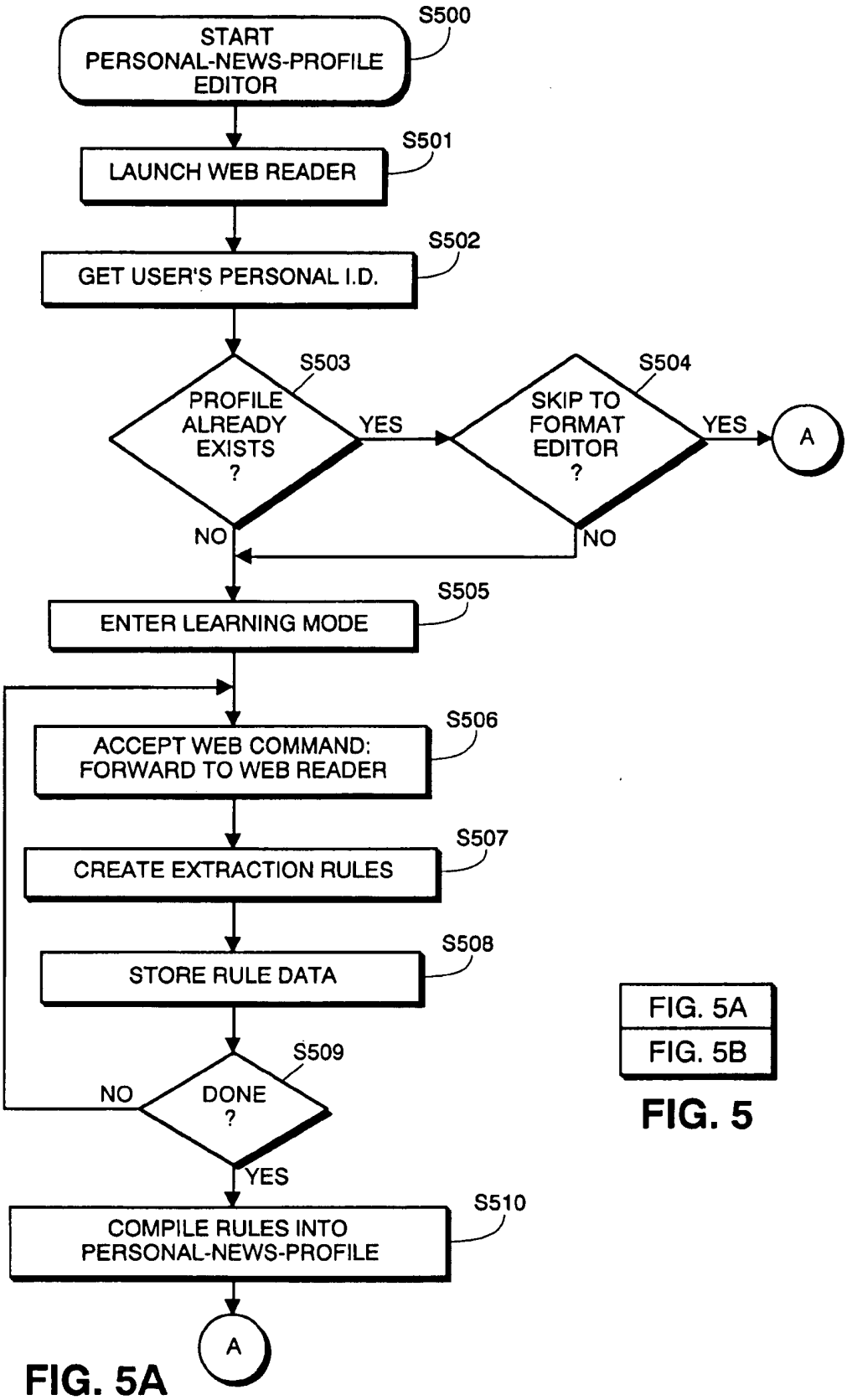


FIG. 5A

FIG. 5A
FIG. 5B

FIG. 5

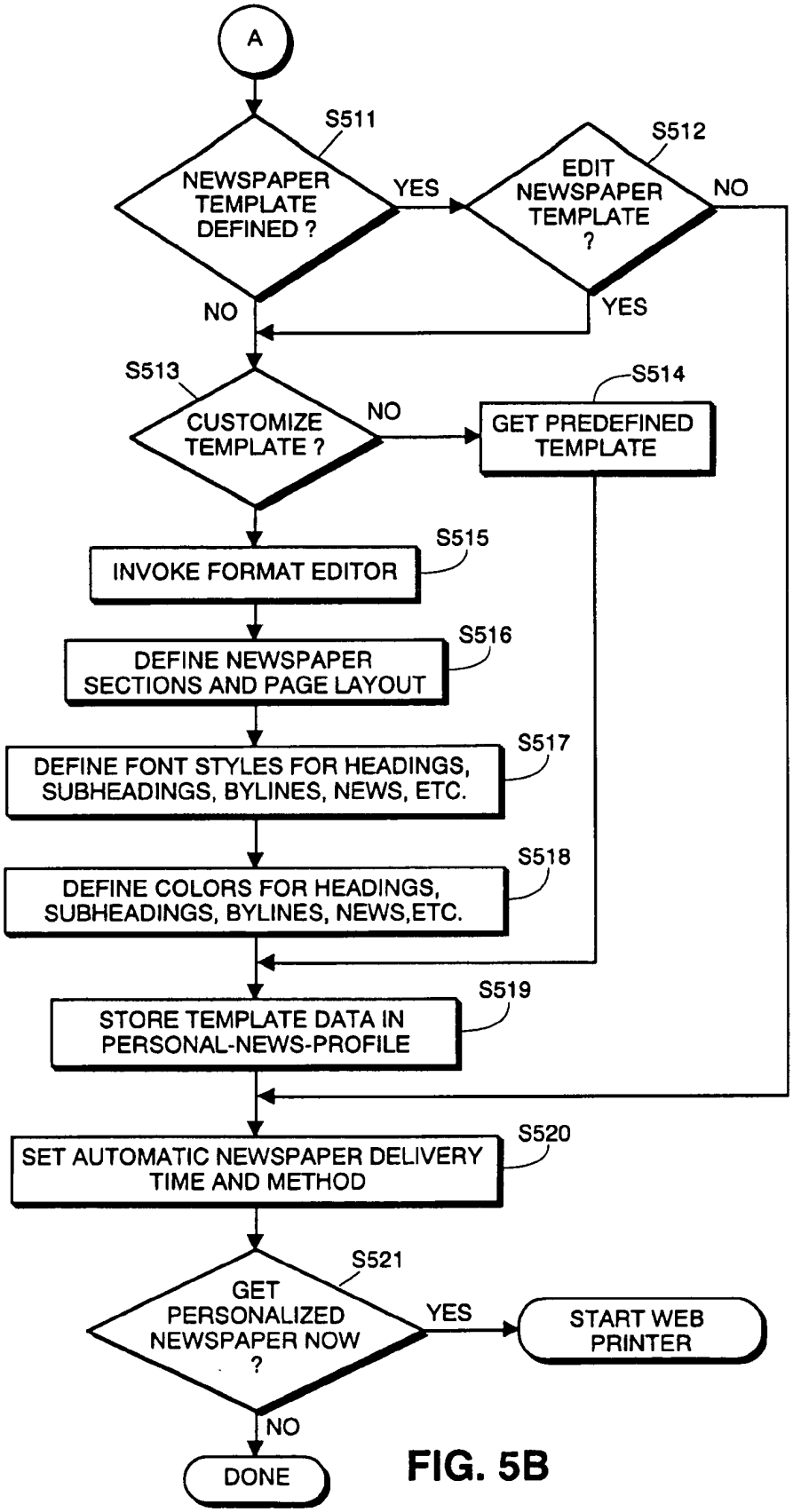

A
S511
NEWSPAPER TEMPLATE DEFINED ?
YES
S512
EDIT NEWSPAPER TEMPLATE ?
NO
YES
NO
S513
CUSTOMIZE TEMPLATE ?
NO
S514
GET PREDEFINED TEMPLATE
S515
INVOKE FORMAT EDITOR
S516
DEFINE NEWSPAPER SECTIONS AND PAGE LAYOUT
S517
DEFINE FONT STYLES FOR HEADINGS, SUBHEADINGS, BYLINES, NEWS, ETC.
S518
DEFINE COLORS FOR HEADINGS, SUBHEADINGS, BYLINES, NEWS,ETC.
S519
STORE TEMPLATE DATA IN PERSONAL-NEWS-PROFILE
S520
SET AUTOMATIC NEWSPAPER DELIVERY TIME AND METHOD
S521
GET PERSONALIZED NEWSPAPER NOW ?
YES
START WEB PRINTER
NO
DONE

FIG. 5B

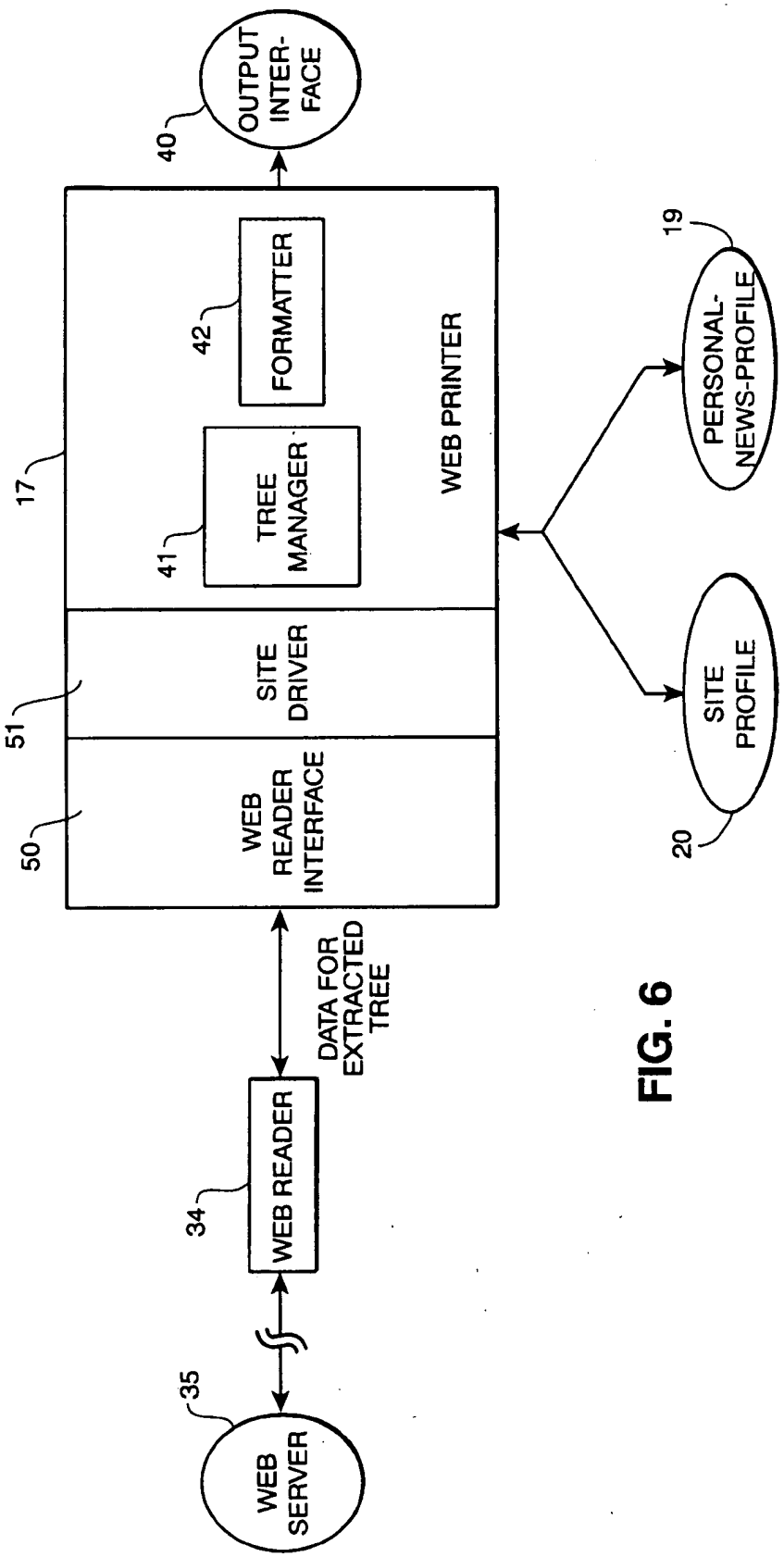

OUTPUT INTER-FACE
40
FORMATTER
42
TREE MANAGER
41
WEB PRINTER
17
SITE DRIVER
51
WEB READER INTERFACE
50
PERSONAL-NEWS-PROFILE
19
SITE PROFILE
20
DATA FOR EXTRACTED TREE
WEB READER
34
WEB SERVER
35

FIG. 6

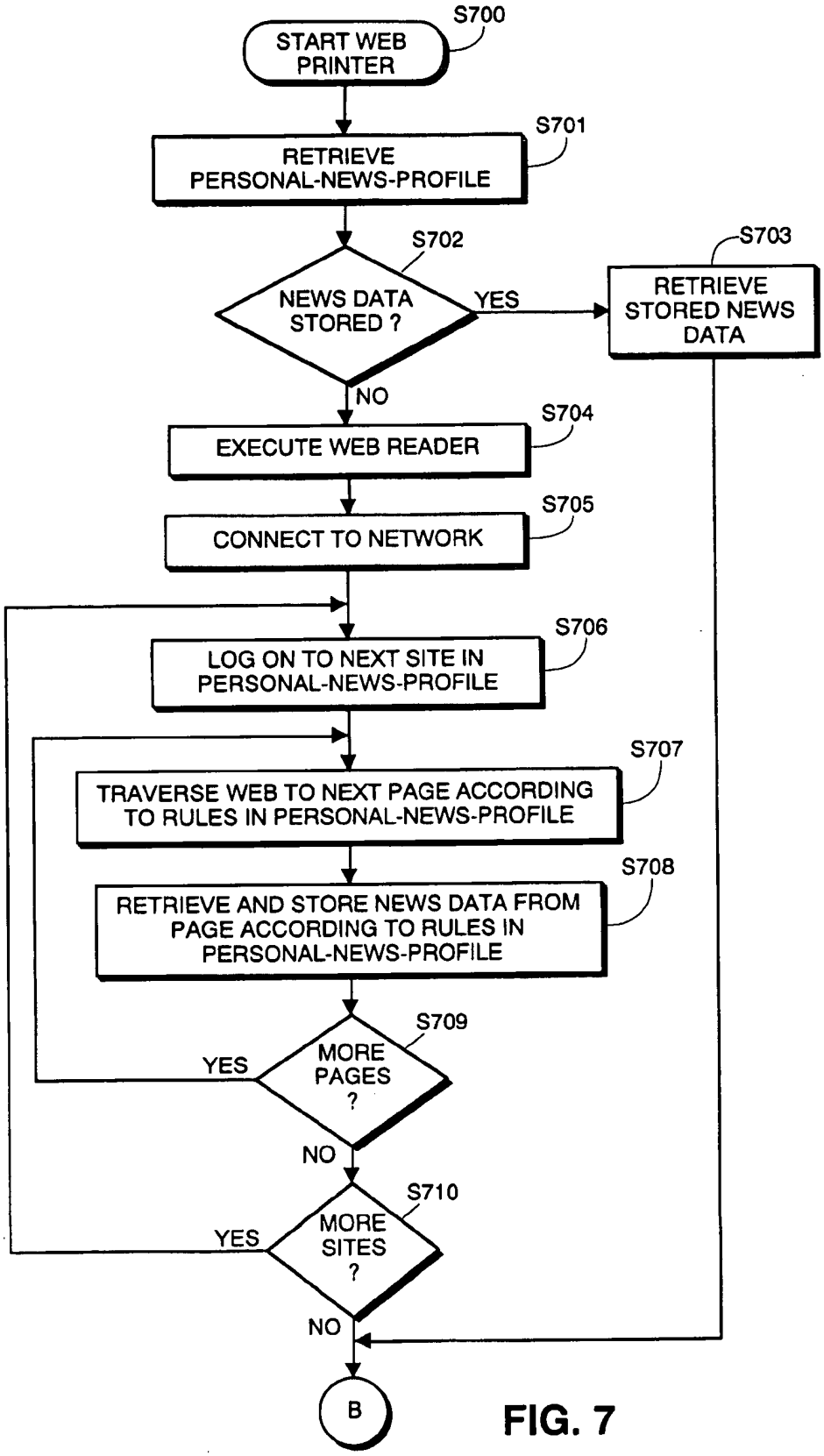

S700
START WEB PRINTER
S701
RETRIEVE PERSONAL-NEWS-PROFILE
S702
NEWS DATA STORED ?
YES
S703
RETRIEVE STORED NEWS DATA
NO
S704
EXECUTE WEB READER
S705
CONNECT TO NETWORK
S706
LOG ON TO NEXT SITE IN PERSONAL-NEWS-PROFILE
S707
TRAVERSE WEB TO NEXT PAGE ACCORDING TO RULES IN PERSONAL-NEWS-PROFILE
S708
RETRIEVE AND STORE NEWS DATA FROM PAGE ACCORDING TO RULES IN PERSONAL-NEWS-PROFILE
S709
MORE PAGES ?
YES
NO
S710
MORE SITES ?
YES
NO
B

FIG. 7

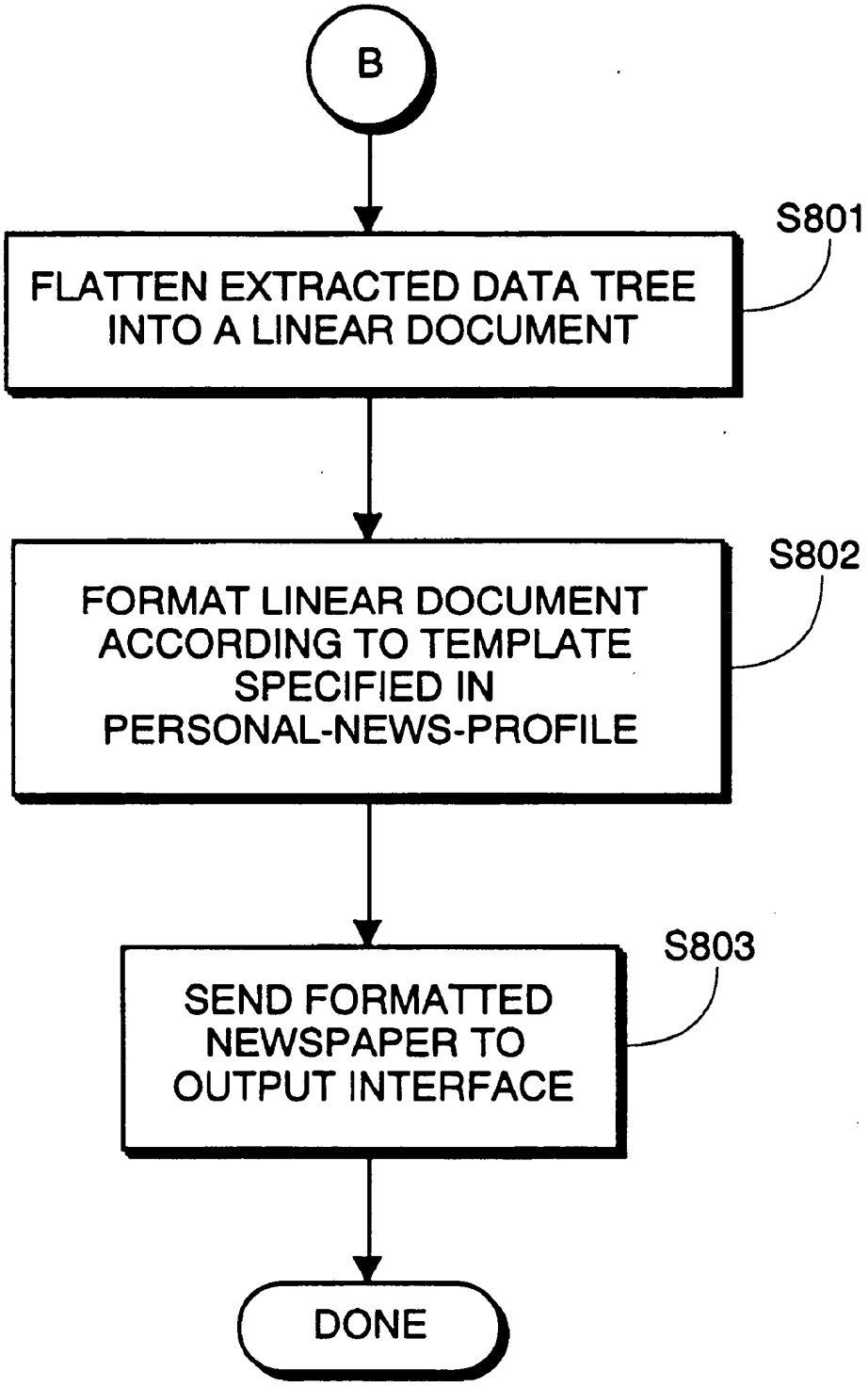

B
S801
FLATTEN EXTRACTED DATA TREE INTO A LINEAR DOCUMENT
S802
FORMAT LINEAR DOCUMENT ACCORDING TO TEMPLATE SPECIFIED IN PERSONAL-NEWS-PROFILE
S803
SEND FORMATTED NEWSPAPER TO OUTPUT INTERFACE
DONE

FIG. 8

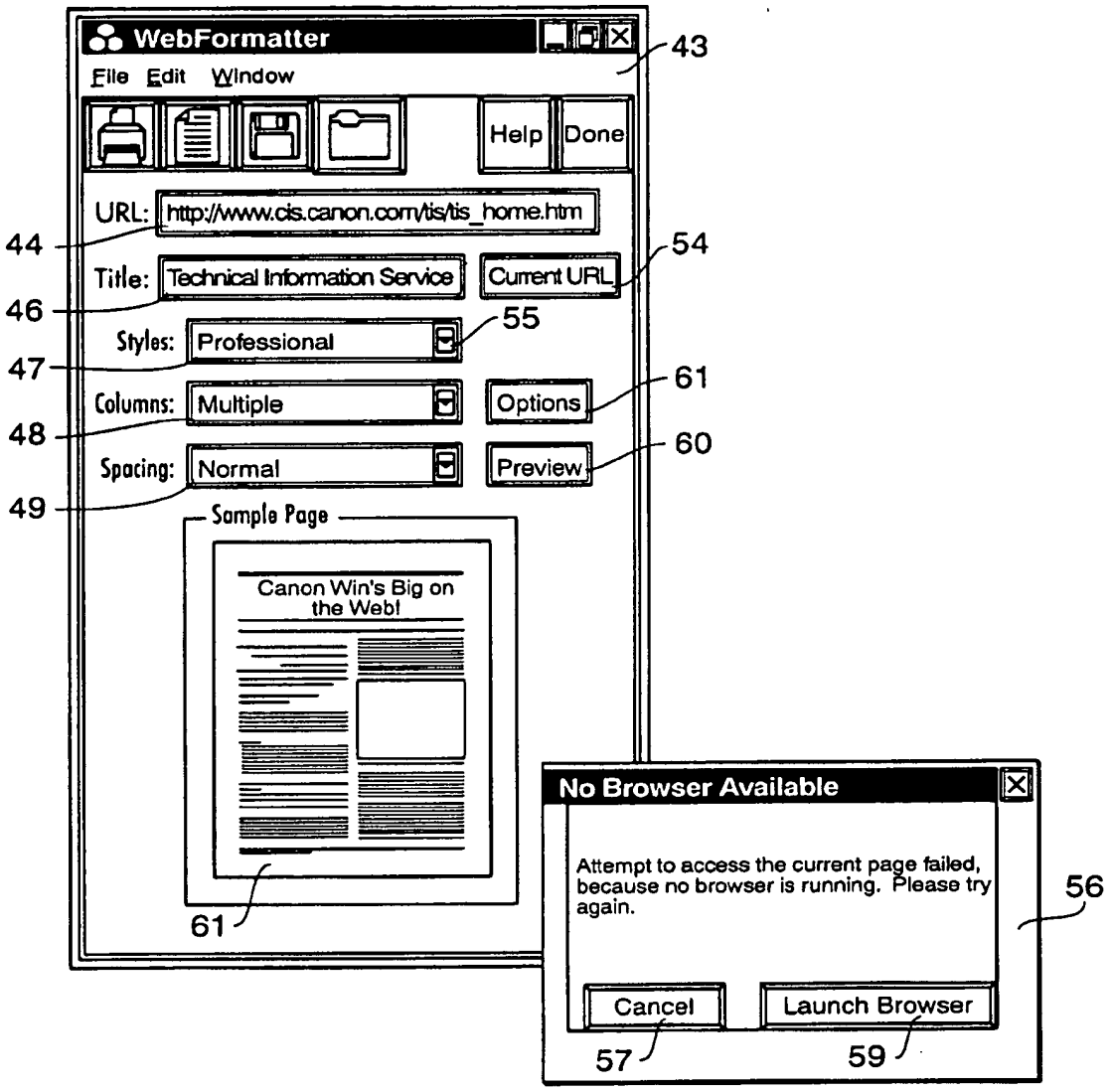

WebFormatter
File Edit Window
Help
Done
URL:
http://www.cis.canon.com/tis/tis_home.htm
Title:
Technical Information Service
Current URL
Styles:
Professional
Columns:
Multiple
Options
Spacing:
Normal
Preview
Sample Page
Canon Win's Big on the Web!
No Browser Available
Attempt to access the current page failed, because no browser is running. Please try again.
Cancel
Launch Browser
43
44
46
47
48
49
54
55
60
61
56
57
59

FIG.9A

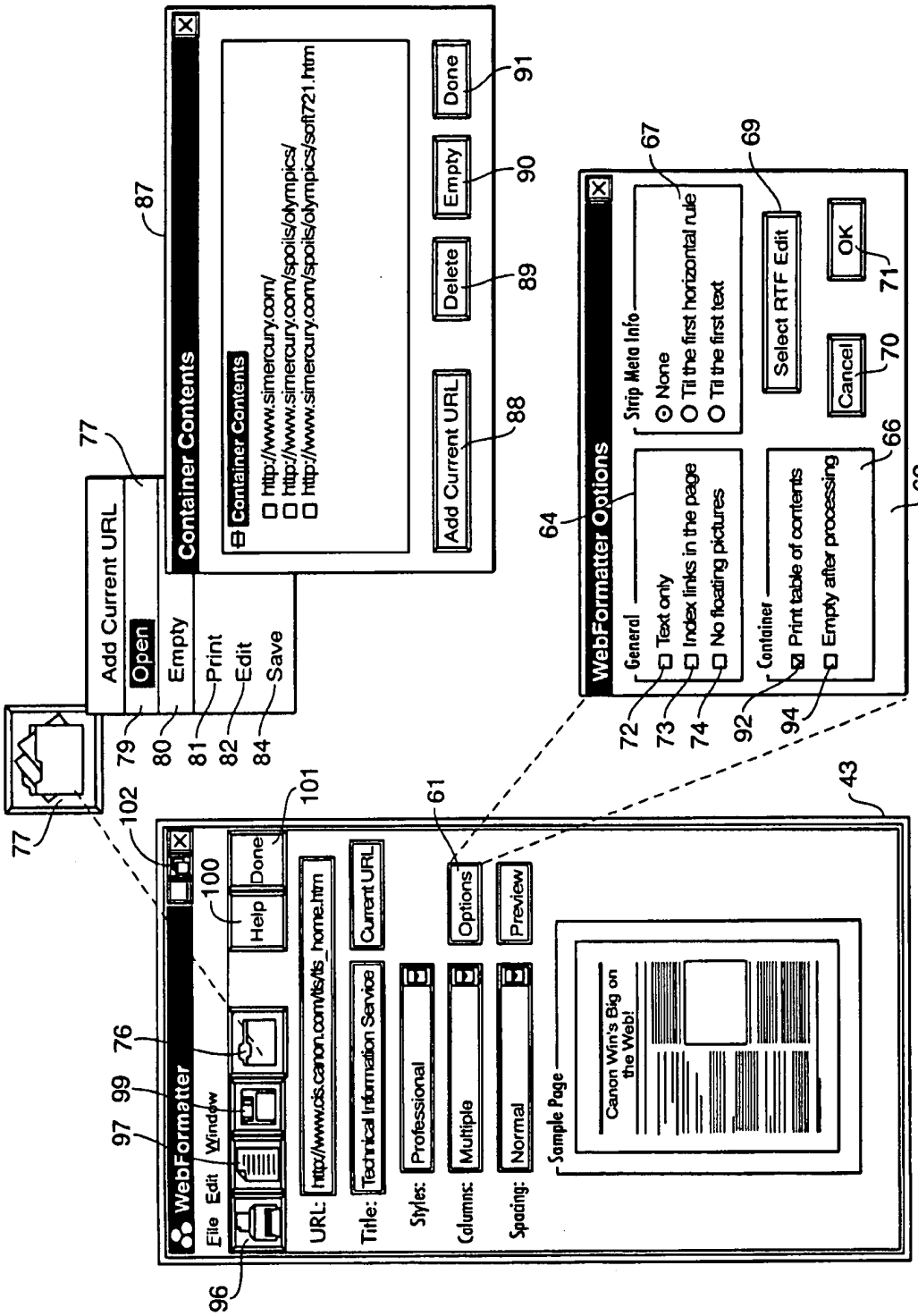

WebFormatter
File Edit Window
URL: http://www.cis.canon.com/tis/tis_home.htm
Title: Technical Information Service
Styles: Professional
Columns: Multiple
Spacing: Normal
Sample Page
Canon Win's Big on the Web!
Done
Help
Current URL
Options
Preview
Add Current URL
Open
Empty
Print
Edit
Save
Container Contents
Container Contents
http://www.simercury.com/
http://www.simercury.com/spoils/olympics/
http://www.simercury.com/spoils/olympics/soft721.htm
Add Current URL
Delete
Empty
Done
WebFormatter Options
General
Text only
Index links in the page
No floating pictures
Strip Meta Info
None
Til the first horizontal rule
Til the first text
Container
Print table of contents
Empty after processing
Select RTF Edit
Cancel
OK
43
61
62
64
66
67
69
70
71
72
73
74
77
79
80
81
82
84
87
88
89
90
91
92
94
96
97
99
100
101
102

FIG.9B

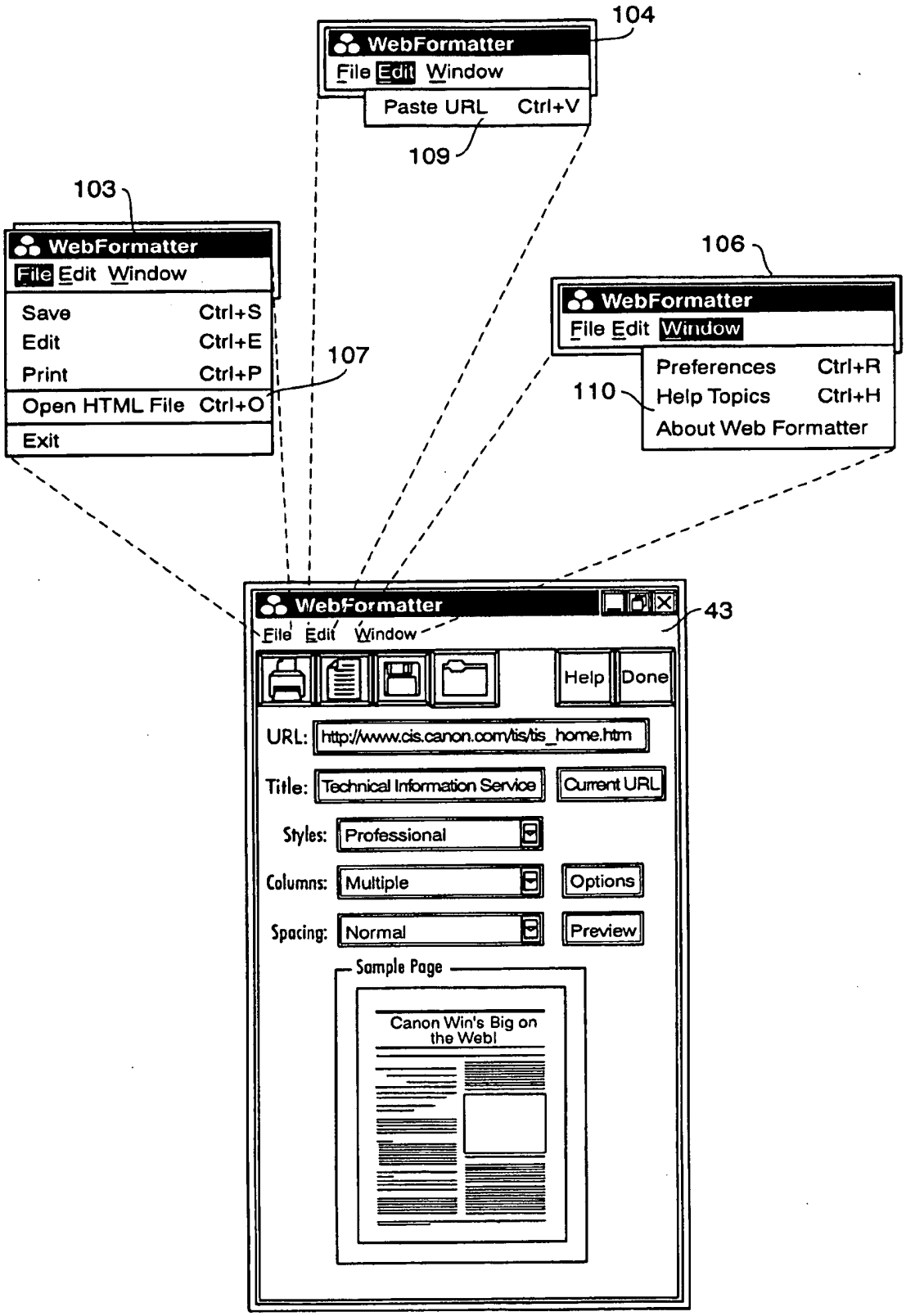

104
WebFormatter
File Edit Window
Paste URL Ctrl+V
109
103
WebFormatter
File Edit Window
Save Ctrl+S
Edit Ctrl+E
Print Ctrl+P
Open HTML File Ctrl+O
Exit
107
106
WebFormatter
File Edit Window
Preferences Ctrl+R
Help Topics Ctrl+H
About Web Formatter
110
WebFormatter
File Edit Window
43
Help
Done
URL: http://www.cis.canon.com/tis/tis_home.htm
Title: Technical Information Service
Current URL
Styles: Professional
Columns: Multiple
Options
Spacing: Normal
Preview
Sample Page
Canon Win's Big on the Web!

FIG.9C

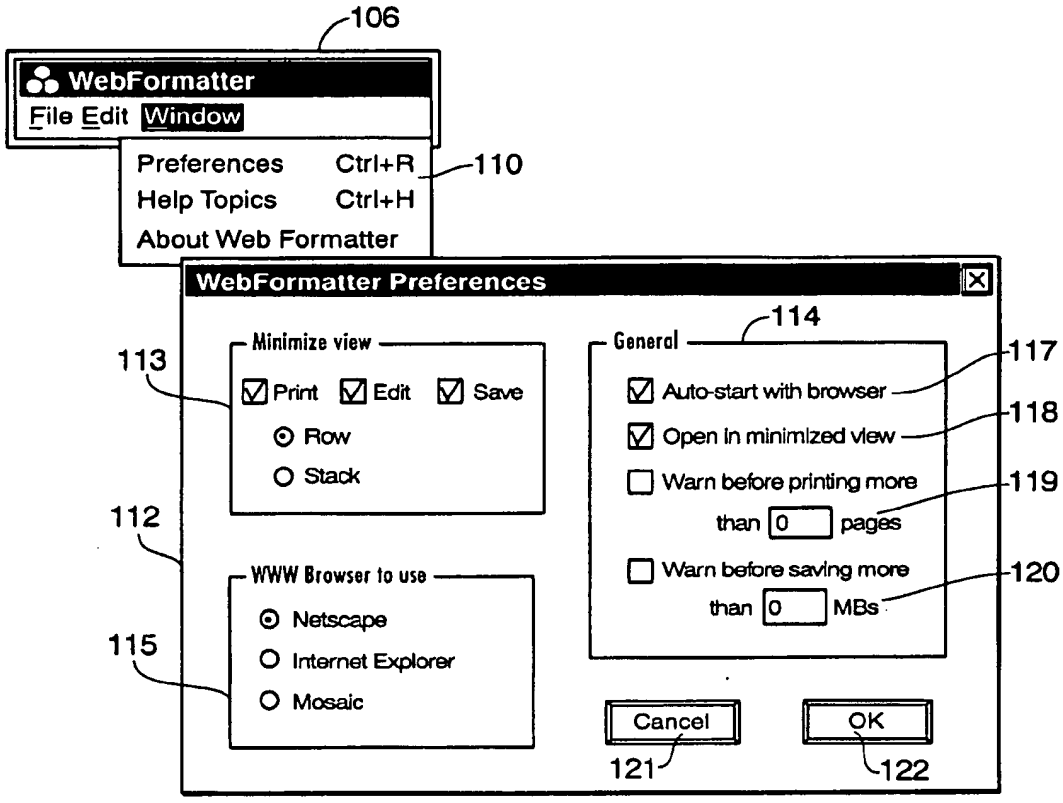

106
WebFormatter
File Edit Window
Preferences Ctrl+R
Help Topics Ctrl+H
About Web Formatter
110
WebFormatter Preferences
113
Minimize view
Print Edit Save
Row
Stack
112
WWW Browser to use
Netscape
Internet Explorer
Mosaic
115
114
General
117
Auto-start with browser
118
Open in minimized view
119
Warn before printing more
than 0 pages
120
Warn before saving more
than 0 MBs
Cancel
OK
121
122

FIG.9D

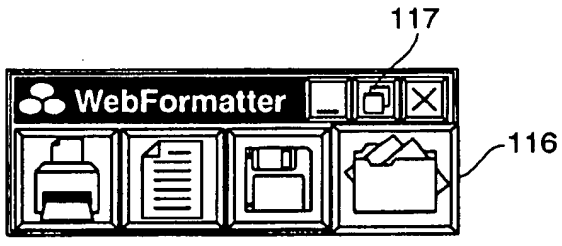

117
WebFormatter
116

FIG.9E

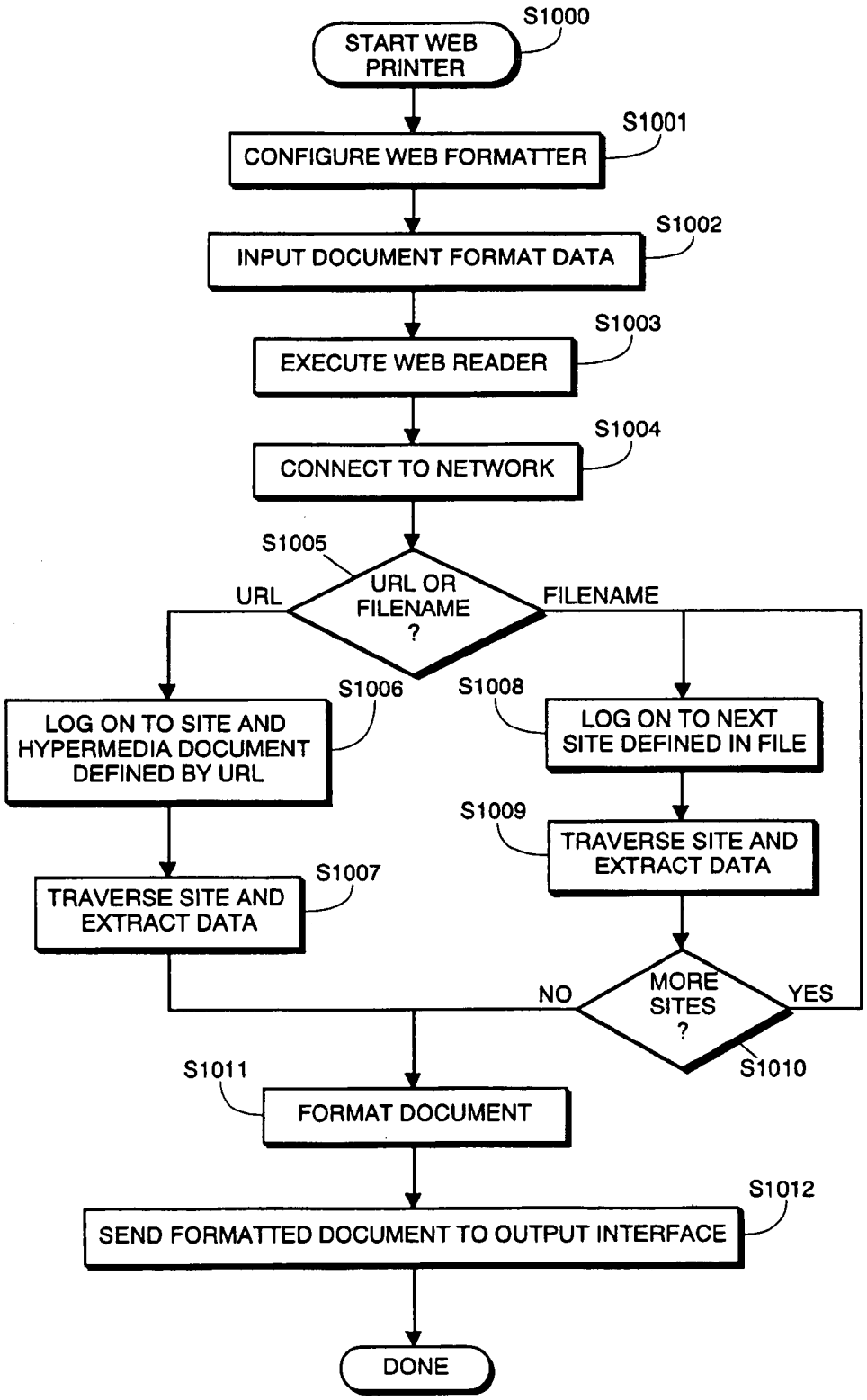

S1000
START WEB PRINTER
S1001
CONFIGURE WEB FORMATTER
S1002
INPUT DOCUMENT FORMAT DATA
S1003
EXECUTE WEB READER
S1004
CONNECT TO NETWORK
S1005
URL OR FILENAME ?
URL
FILENAME
S1006
LOG ON TO SITE AND HYPERMEDIA DOCUMENT DEFINED BY URL
S1007
TRAVERSE SITE AND EXTRACT DATA
S1008
LOG ON TO NEXT SITE DEFINED IN FILE
S1009
TRAVERSE SITE AND EXTRACT DATA
NO
MORE SITES ?
YES
S1010
S1011
FORMAT DOCUMENT
S1012
SEND FORMATTED DOCUMENT TO OUTPUT INTERFACE
DONE

FIG. 10

SYSTEM FOR GENERATING A CUSTOM FORMATTED HYPERTEXT DOCUMENT BY USING A PERSONAL PROFILE TO RETRIEVE HIERARCHICAL DOCUMENTS

BACKGROUND OF THE INVENTION

1. Field of the Invention

The invention relates to a data retrieval system which automatically traverses hypermedia documents on a computer network and automatically retrieves information from those documents based on a match between the structure of the documents and a personalized data retrieval structure. More particularly, the invention can retrieve articles from a news service, from a magazine service, or from a combination of both services which are located on the World Wide Web, a private computer network that supports hypermedia links, or any other hypermedia-linked computer system.

For example, there exists a Web site for retrieving news articles from the New York Times and a Web site for retrieving articles from People magazine. The retrieval system of the invention can traverse through such Web sites and select articles based on a personalized data retrieval structure. The personalized data retrieval structure may include commands to retrieve a full text of the front page only, headlines of the business section, headlines of the stock section and sports section, etc. In addition, the personalized data retrieval structure may include content-based rules to retrieve articles with certain keywords, to exclude articles with certain keywords, or to include articles based on a rule-based content analysis. The invention also provides a method for synthesizing all retrieved news articles and printing the synthesized news articles into a newspaper-type format in which each of the articles is arranged based on a user's predefined layout.

While the above example is in the context of the Web, hypermedia documents can reside on other types of networks besides the Web, such as an intranet. An intranet is a private computer network that is not connected to outside computer networks. For example, a company's own computer network could be an intranet with hypermedia documents on it. For brevity, the following discussion is made with respect to the World Wide Web. However, it should be understood that the invention applies equally well to any type of computer network that contains hypermedia documents, such as an intranet, different hypermedia-linked computer networks that reside on the Internet other than the Web, etc.

A hypermedia document on the Web can span multiple Web sites. Such documents can be newspapers, news articles, magazines, catalogs, manuals, memoranda, and the like. For brevity, the following discussion is made with respect to sources of news information. However, it should be understood that the invention applies equally well to any other type of hypermedia document.

2. Description of the Related Art

The World Wide Web is an on-line source of hypermedia documents containing hypermedia text and images that act as links to other documents, Web sites, etc. As a result, documents on the Web are not organized sequentially. Rather, a user is automatically linked to other documents or Web sites to complete the viewing of a document by selecting a hypermedia link, such as a text link or an image link, within the document. Accordingly, an entire document cannot be viewed by scrolling through text.

One popular use of the Web is on-line publication and distribution of magazines and newspapers. Currently, many

Web news services, such as the New York Times, allow the user to define keywords of interest and to receive news information, daily or hourly, that contains text matching the keywords. The news information can then be delivered to the user's computer via modem or E-mail. However, most Web news site newspapers, like the New York Times, include too much information, most of which has no interest to the user since the information is retrieved based only on a keyword match.

Other sources of news information are provided through information suppliers like "Individual Inc." Individual Inc. supplies users with a brief summary of the top twenty most relevant articles based on a user's predefined keywords. This subscription news service allows the user to specify five to ten areas of interest based on keywords, which are then prioritized by the user. The information service searches the Web for magazines and newspapers which contain any of the keywords. Based on the keyword searches, twenty of the most relevant articles are selected, compiled into a brief one-page summary, and transmitted to the user via facsimile for the user's review. However, in order to review an entire document rather than the summary, the user must log onto a specific Web site containing the document in order to retrieve and review the document.

There are yet other services which permit the user to personalize a newspaper to be displayed at the user's terminal by storing links to various news articles from various news sources on the Web. For example, CRAYON "Create Your Own Newspaper" permits a user to select specific sections from among links to over twenty-five different on-line newspapers, and to compose the selections into a personalized newspaper. Using CRAYON, it is possible to compose a personalized newspaper containing, for example, links to the international section of the New York Times, the business section of the Wall Street Journal, and the sports section of the Chicago Tribune. The HTML (hypertext markup language) source file for this newspaper is then stored to mass media storage for later use.

While the forgoing news and information services provide convenient ways to keep updated on the news, they do not allow a user to access and view the news in the way that people naturally read a real-world newspaper. Namely, people naturally read a newspaper by scanning the pages of sections that they find interesting and then reading those articles that grab their attention. In other words, people use a structural approach to decide what pages to look at initially (e.g., the first page of the Business and World sections, and the comics page of the Arts section). They then scan the selected pages for articles.

In sum, conventional news and information services do not allow a user to access data from a hypermedia document on the basis of the structure of the document, and then to format that data in a manner that allows the user to scan and read the data in a natural fashion.

SUMMARY OF THE INVENTION

The invention addresses the above deficiencies in the art by accessing at least one hypermedia document, retrieving data from the hypermedia document into an extracted data tree, with the data retrieved based on a structure of the hypermedia document, flattening the extracted data tree into a linear document, and formatting the linear document into a formatted document.

In another aspect, the invention creates a personal-news-profile for retrieving data from a hypermedia-linked computer network. The hypermedia-linked computer network is

accessed, a learning mode is started, the hypermedia-linked computer network is traversed with commands, at least one rule is extracted from the commands, and the rule(s) is compiled into the personal-news-profile.

In yet another aspect, the invention creates a personalization profile for a Web site retrieval data retrieval system. Data and commands are input to access the World Wide Web and a connection is made to the World Wide Web. A Web reader is launched, and the Web reader accesses the Web via the connection. In response to user commands, a learning mode is entered into. Commands are sent to traverse the World Wide Web, and at least one rule is extracted from the commands. The rule(s) is compiled into a personalization profile, which is stored.

In yet another aspect, the invention retrieves articles from a hypermedia-linked computer network and formats the articles into a personalized newspaper. A stored personal-news-profile is retrieved. The personal-news-profile includes address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands. The site is accessed based on address data stored in the personal-news-profile, and articles at the site are downloaded based on command data stored in the personal-news-profile. The downloaded articles are flattened into a linear document, and the linear document is formatted into the personalized newspaper according to newspaper layout commands stored in the personal-news-profile.

In yet another aspect, the invention retrieves data from a World Wide Web site and formats the data into a personalized document. A Web site data retrieval driver which includes a Web reader, stored Web site address information, stored Web site commands, and stored format information is accessed. The invention (1) launches the Web reader to connect to the World Wide Web via a connection to the Web, (2) retrieves the Web site address information and Web site commands, (3) instructs the Web reader to access the Web site based on the Web site address information and Web site commands, (4) downloads Web site data from the Web site based on the Web site commands, wherein the data is downloaded with reference to a linked list so as to avoid hypermedia-links that form loops and so as to avoid repetitious downloading of data that has already been downloaded, (5) stores the Web site data in a linear document, (6) repeats steps 2 through 5 until all addresses in the stored Web site address information have been accessed, and (7) formats the linear document into the personalized document based on the format information.

In yet another aspect, the invention accesses and retrieves data at World Wide Web sites and formats the data into a personalized document. The invention connects to the World Wide Web, retrieves user defined Web site address information, user defined Web site commands, and user defined formatting commands, and activates a Web reader so as to access a Web site based on the user defined Web site address information. The Web reader is used to download data from the Web based on the user defined Web site commands, and the data is downloaded into an extracted data tree. The downloading continues until all addresses in the user defined Web site address information have been accessed. The extracted data tree is flattened into a linear document, and the flattened document is formatted into the personalized document based on the user defined formatting commands.

In yet another aspect, the invention retrieves news articles from on-line news services on the World Wide Web and formats the news articles into a personalized newspaper. The invention stores a personal-news-profile which comprises addresses data and command data for accessing data from a Web site and newspaper format commands, retrieves the stored personal-news-profile and accesses the data stored therein, activates a Web reader to contact a Web site based on address data stored in the personal-news-profile, downloads news articles at the contacted Web site based on command data stored in the personal-news-profile, stores the downloaded news articles, and formats the stored news articles into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile.

In yet another aspect, the invention formats a hypermedia document into a personalized document. A location of the hypermedia document is specified, a type of the hypermedia document is specified, a scope of data to be retrieved from the hypermedia document is specified, wherein the scope is based on a structure of the hypermedia document, and a format is specified for formatting the data retrieved from the hypermedia document into the personalized document. The hypermedia document found at the specified location is accessed, data is retrieved from the hypermedia document in accordance with the specified hypermedia document type and in accordance with the specified scope, and the data is formatted into the personalized document in accordance with the specified format.

In yet another aspect, the invention is a system for processing a hypermedia document. The system accesses the hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from downloaded data in accordance with predetermined configuration information. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

In preferred embodiments, the system inputs the formatting settings and configuration information via a graphical user interface. The graphical user interface comprises plural processing icons, one of which activates the processing function. By virtue of the graphical user interface, a user can interactively set a document's format and change that format should a change be desired.

In particularly preferred embodiments, the graphical user interface is displayed in plural modes. The plural modes comprise (1) a fully-functional mode in which the graphical user interface displays formatting fields, processing options, menus and the processing icons, and (2) a minimizing mode in which the graphical user interface displays only the processing icons. Typically, the graphical user interface displayed in the minimizing mode is displayed during browsing the hypermedia document. By displaying the graphical user interface in plural modes, the present invention facilitates operation of the invention during browsing of the hypermedia document.

This summary has been provided so that the nature of the invention may be understood quickly. A more complete understanding of the invention can be obtained by reference to the following detailed description of the preferred embodiments thereof in connection with the attached drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. 1 is a perspective view showing the outward appearance of the personal news retrieval system according to the invention.

FIG. 2 is a block diagram of the personal news retrieval system shown in FIG. 1.

FIG. 3, comprised of FIGS. 3A, 3B, 3C and 3D, shows representational diagrams illustrating an example of the transformation of information from the Web (FIG. 3A) to an extracted data tree (FIG. 3B), then to a flattened document (FIG. 3C), and finally to a formatted document (FIG. 3D) according to the invention.

FIG. 4 is a representational block diagram of the manner by which a personal-news-profile for retrieving news articles via the Web is created or edited according to the invention.

FIG. 5, comprised of FIGS. 5A and 5B, shows flow diagrams describing how a personal-news-profile is created or edited.

FIG. 6 is a representational block diagram of the manner by which news articles are retrieved from the Web and formatted with reference to a personal-new-profile according to the invention.

FIG. 7 is a flow diagram describing how news articles are retrieved from the Web with reference to a personal-news-profile.

FIG. 8 is a flow diagram showing how retrieved news articles are formatted with reference to a personal news profile and sent to a print device interface.

FIGS. 9A to 9E depict a graphical user interface used with the second embodiment of the present invention.

FIG. 10 is a flow diagram describing the operation of the second embodiment of the invention.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENT

FIG. 1 is a view showing the outward appearance of a representative embodiment of the invention. Shown in FIG. 1 is computing equipment 1, such as a MacIntosh or an IBM PC or a PC-compatible computer, having a windowing environment, such as Microsoft Windows. Provided with computing equipment 1 is display screen 2, such as a color monitor or a monochromatic monitor, keyboard 3 for entering text data and user commands, and a pointing device such as mouse 4 for pointing and for manipulating objects displayed on display 2. Computing equipment 1 also includes a mass storage device such as disk drive 5. Image data can be input into computing equipment 1 from a variety of sources such as a network interface 11*a* or from external devices via facsimile/modem interface 6. Network interface 11*a* is used to connect computing equipment 1 to a local area network (LAN) or to a wide area network (WAN) such as the World Wide Web.

FIG. 2 is a detailed block diagram showing the internal construction of computing equipment 1. As shown in FIG. 2, computing equipment 1 includes central processing unit (CPU) 8 interfaced with computer bus 9. Also interfaced with computer bus 9 is printer interface 10, fax/modem interface 6, display interface 11, network interface 11*a*, keyboard interface 12, mouse interface 13, main memory 14, and disk drive 5.

Main memory 14 interfaces with computer bus 9 so as to provide random access memory storage for use by CPU 8 when executing an application such as personal-news-profile editor 16 or Web printer 17. More specifically, CPU 8 loads these software applications from disk drive 5 into main memory 14 and executes the software applications out of main memory 14. In accordance with user instructions, stored application programs are activated which permit processing and manipulation of data. Typically, the software applications stored on disk drive 5, such as personal-news-profile editor 16, Web printer 17, and HTML formatter 18, have been stored on disk drive 5 by downloading the software applications from a computer-readable medium such as a floppy disk or CD ROM, or by downloading the software applications from a computer bulletin board.

Disk drive 5 stores data files which can include text files and image files, in compressed or uncompressed format, and stores software application files such as those noted above. The software application files include Windows applications, DOS application, and personal news retrieval files 15. Personal news retrieval files 15 include personal-news-profile editor 16, Web printer 17, HTML formatter 18, personal-news-profile(s) 19, and site profile(s) 20. The detailed functions of personal news retrieval files 15 will be discussed below, after a brief overview of the operation of the personal new retrieval system.

Overview of Document Retrieval

FIG. 3, comprised of FIGS. 3A to 3D, illustrates the operation of a representative embodiment of the invention. FIG. 3A is a graphical representation of a typical Web site 21 with news information contained therein. Within Web site 21 is homepage 22 with links to indices such as headings 23, which are in turn linked to articles 24. Some of articles 24 are linked to other articles. As article H 26 resides on another Web site, link 25 is a cross-site link. Link 25 illustrates how a single hypermedia document, represented by homepage 22, can traverse multiple Web sites.

In order to retrieve news from Web site 21, the invention first traverses Web site 21 to retrieve data according to user-defined rules. As will be discussed in more detail below, these rules can be based on the structure of Web site 21, or on the structure of Web site 21 and its contents. The data is retrieved into an extracted data tree, which preserves the organization of the data as shown in FIG. 3B, but in which some links are excluded.

The organization of extracted data tree 27 has several features. First, extracted data tree 27 has root 28 which can have child nodes for one or more sites 29, which in turn can have index nodes 30 which correspond to indices/headings 23, articles nodes 31, and the like. Second, extracted data tree 27 is a true tree, with no loops (i.e., cyclic paths) therein. For example, FIG. 3A shows a loop from homepage 22 to index node #1, to article C, and then back to homepage 22. This loop is removed when creating extracted data tree 27.

Second, the organization of extracted data tree 27 depends on how the Web sites are traversed, and not on the Web sites' actual layouts. Thus, article H 26 appears under index node #3 (under site #1), indicating that the news retrieval system accessed article H 26 from site #1 via cross-site link 25.

Finally, as noted earlier, certain articles have been excluded from extracted data tree 27 due to the structure of Web site 21 or possibly a content of indices/headings 23 and articles 24. For example, articles E and G have been excluded from extracted data tree 27.

According to the invention, extracted data tree 27 is flattened into linear document 32, as shown in FIG. 3C, possibly with reference to more exclusion rules. Linear document 32 is simply a continuous document with information from extracted data tree 27 embedded therein.

Finally, linear document 32 is formatted according to user-specified (or default) formatting instructions into formatted document 33, shown as a stylized personal newspaper in FIG. 3D. Formatted document 33 has various fonts and/or colors for site labels, indices/headings, articles, and

the like. Furthermore, formatted document **33** is broken down into pages.

Note that in alternate embodiments of the news retrieval system, certain stages of the above transformation from Web site **21** to formatted document **33** can be skipped. For example, data from Web site **21** can be retrieved directly into flattened document **32**, as long as a record of the organization of the data is maintained (possibly in a separate linked list) so as to avoid downloading the same article twice and so as to avoid loops in the organization of Web site **21**. Alternatively, extracted data tree **27** can be directly formatted into formatted document **33**. In any case, the basic operation of the invention remains the same: the news retrieval system traverses a hypermedia document on the Web, extracts data according to user-defined information, and formats the data into a personalized newspaper.

As mentioned in the above discussion, various user-defined rules and other information (such as formatting information) are involved in the news retrieval process. That user defined information is stored in personal-news-profile(s) **19**, the definition of which is described next.

Defining a Personal-News-Profile

FIGS. **4** and **5** illustrate the process by which personal-news-profile **19** is defined. To create personal-news-profile **19**, personal-news-profile editor **16** communicates with personal-news-profile **19**, site profile **20**, and Web reader **34**.

Personal-news-profile **19** contains information as to what sites to access for creating a personalized newspaper, what sections to retrieve from those sites, rules to be used to determine what data to extract from the sections and the articles therein, rules to determine how to exclude links, and newspaper format information. A sample personal-news-profile is shown in Appendix 1.

Site profile **20** includes general site information that is not specific to a particular user. For example, site profile **20** could contain information such as full site addresses, sections within a site, non-user specific passwords, etc. Sample site profiles are shown in Appendix 1. Because general site information is stored in site profile **20**, personal-news-profile **19** can refer to the general site information with reference to site profile **20**, saving space in the personal-news-profile. For example, as shown in Appendix 1, personal-news-profile **19** can refer to a site number **1**. Site profile **20** indicates that site number **1** is the "San Jose Mercury News," with a homepage at "http//www.sjmercury.com/". This construction also centralizes general site information. Thus, if a site address changes, only site profile **20** needs to be changed to update all personal-news-profiles **19** on the system.

Web reader **34** is an application program or program module that communicates with the Web via Web server **35**. In response to commands from personal-news-profile editor **16**, Web reader **34** will access the Web, traverse hypermedia documents on the Web, retrieve data from the documents, and return the retrieved data to personal-news-profile editor **16**.

As shown in FIG. **4**, personal-news-profile editor **16** includes four modules: site driver **36**, Web reader interface **37**, profile manager **38**, and format editor **39**.

Web reader interface **37** interfaces personal-news-profile editor **16** to Web reader **34**. Site driver **36** interacts with Web reader **34** via Web reader interface **37** to provide an abstract interface to each individual Web site. More specifically, site driver **36** instructs Web reader **34** to access various Web sites and to retrieve data from those sites. Thereafter, site driver **36** receives that data and builds site profile **20** therefrom. The data can also be used to update an existing site profile.

In building site profile **20**, site driver **36** translates the structure of each accessed Web site to a uniform structure defined in site profile **20**, and stores data retrieved therefrom in site profile **20**. By translating different Web sites, some of which may have different structures, into a single uniform structure and storing data therefrom in that structure in site profile **20**, the present invention facilitates access to information from different Web sites, and thus reduces overall processing time.

Profile manager **38** maintains document templates that specify how to format a personalized newspaper. Predefined document templates exist. In addition, format editor **39** allows a user to specify personalized templates for formatting a newspaper, either by editing existing templates or by creating new ones. In any case, each document template specifies page layout information, font information, style information, colors, etc. for the titles, indices/headings, subheadings, text and the like for a personalized newspaper.

Sample code for personal-news-profile editor **16**, site driver **36**, and profile manager **38** is included in Appendix 3A.

FIGS. **5A** and **5B** are flow diagrams describing the operation of personal-news-profile editor **16** in more detail. FIG. **5A** shows the operation of personal-news-profile editor **16** in defining the parts of personal-news-profile **19** relating to accessing Web sites and retrieving data from those sites.

In step **S500** of FIG. **5A**, personal-news-profile editor **16** is launched by a user. In step **S501**, the editor launches Web reader **34**. The user's personal I.D. is then retrieved in step **S502**. If a personal-news-profile already exists for that I.D., step **S503** directs flow to step **S504**, where the user is given the option of skipping to the format editor. Otherwise, personal-news-profile editor **16** enters a "learning mode" in step **S505**. Once in the learning mode, personal-news-profile editor **16** proceeds to step **S506**, where it accepts a Web command (i.e., a command to traverse a hypermedia link) from the user and forwards the Web command to the Web reader by means of site driver **36**. Site driver **36** maintains a hierarchical log of Web sites visited by Web reader **34**. In step **S507**, personal-news-profile editor **16** creates an extraction rule from the Web command. This rule will allow the news retrieval system to later duplicate the user's selection criteria in browsing (clicking on hyperlinks within) a Web site.

The rule specifies, at the least, structural criteria for duplicating the traversal of the Web site. For example, if a user accesses all articles under a particular index/heading, the rule will specify that all articles under that index/heading should be retrieved.

In one embodiment of the invention, the rule can also include content-based criteria (i.e., keyword-based criteria) accepted from the user. These content-based rules can, for example: (1) require certain words to be in an article, (2) exclude articles with certain words, (3) require certain boolean combinations of words, (4) rank articles that are selected based on structural criteria, with the ranking based on keywords, and then require the selection of the articles with the highest ranking(s), or (5) exclude certain types of articles such as advertisements.

Examples of the syntax for the structural and content-based exclusion rules are shown in Appendix 2. Several different types of rules are shown. Some simply limit the traversal of a Web site to a certain number of links. Others are date and keyword based exclusion rules. One particularly flexible rule indicates that articles should be ranked based on a keyword analysis and the top scoring articles should be

chosen. Other rules include "flattening" rules. These rules control the flattening of the extracted data tree, as will be explained in more detail below.

At the least, the rule includes structural information about the user's selection (i.e., first page, first document, all links, etc.), necessary password information, browser commands, and the like. The rule can also include a pointer or a reference to site profile **20** and the appropriate information therein. General (non-user specific) information is used by site driver **36** to maintain site profile **20**. In this manner, address information and passwords common to multiple users can be maintained in site profile **20**, as discussed above. For example, site driver **36** will store commands or hyperlinks to other documents in a Web page in the rule, but will not store a Web site's full address in the rule. That address information is stored in site profile **20**.

In step **S508**, rule data defining the rule created from a Web command(s) is stored in an extracted data tree such as extracted data tree **27** in FIG. **3**B. This data tree is a linked list that reflects the organization of the data retrieved from the Web. In step **S509**, flow returns to step **S506** for the next Web command unless the user is done (i.e., the user signs-off the Web site), in which case flow proceeds to step **S510**.

At this point, the creation of the personal-news-profile has proceeded much like the creation of a macro common to word processing programs, except that site profile **20** has been used to minimize storage requirements and to centralize general site information. In order to minimize storage requirements further and in order to make the news retrieval system more flexible and efficient, the extracted rules are now compiled to remove redundant links, multiple visits to the same site, and the like. This occurs in step **S510**, and the resulting compiled rules become the first part of personal-news-profile **19**.

Alternatively, personal-news-profile editor **16** may be invoked as a graphical user interface which allows a user to edit a previously stored personal-news-profile or to specify document composition preferences, for example, by specifying news sites, headline articles only, keywords, etc. In either case, the result is personal-news-profile **19**, which comprises a listing of Web site pointers as well as extracted rules for traversing through a Web site or sites.

For a better understanding of the above, sample personal-news-profiles and sample site profiles are provided in Appendix 1 as noted above.

Next, operation proceeds to give the user an option to modify a custom newspaper template, as shown in FIG. **5**B. In step **S511**, it is determined if a newspaper template has been defined and stored in personal-news-profile **19**. If a newspaper template has been defined, step **S512** gives the user the option to edit the template or to proceed to step **S520**. If the user chooses to edit the template or if no newspaper template has been defined, flow proceeds to step **S513**.

Step **S513** gives the user the option of creating a custom template or using a predefined template. If the user wants to use a predefined template, step **S514** gets the specified predefined template, which is added to the personal-news-profile in step **S519**. Otherwise, flow proceeds to step **S515**, where format editor **39** is invoked.

Format editor **39** has a graphical user interface that provides the user with a number of formatting options. In step **S516**, format editor **39** allows the user to define which newspaper sections are to be printed in the newspaper, which Web site's news article are to be placed in each section, and/or how each page is to be laid out. In this regard, the user can specify which Web site's news articles are to be used as a front page, which Web site's news articles are to be used as a business page, which Web site's news articles are to be used as a sports page, etc. In addition, in step **S516**, the user can define where each index/heading should be listed, as well as what sub-headings should go on each page.

In step **S517**, format editor **39** allows the user to define the font styles for indices/headings, sub-headings, bylines and actual text of news articles. In step **S518**, format editor **39** prompts the user to define index/heading colors, title colors, etc. In this regard, layout editor **39** is capable of determining the types of fonts and colors available to the user based on the system's printer capabilities.

Once all of the information is gathered for the custom template, the format editor adds the information to personal-news-profile **19** in step **S519**. Alternatively, profile manager **38** may also store the custom format as a template in a common area for use by other users. In this case, only a pointer or reference to the custom template is stored in personal-news-profile **19**.

In step **S520**, personal-news-profile editor **16** prompts the user to set an automatic newspaper delivery time and method (i.e., print or store on disk drive **5** for later printing). These settings are added to personal-news-profile **19**. More specifically, in the case that a user's computer is continuously supplied with power, the Web news retrieval system can be launched automatically at a designated time. The system will retrieve articles from the Web sites which are listed in personal-news-profile **19**. Upon retrieving the news articles, the articles will be formatted based on the newspaper template in personal-news-profile **19**. The formatted personalized newspaper can then be either printed or stored for later viewing. In the case that a time is not set for newspaper delivery, the user can execute the Web news retrieval system program at any time.

Once personal-news-profile **19** has been created, the Web news retrieval system, upon being launched, can traverse Web news sites and build a personalized newspaper by automatically retrieving various news articles from the Web news sites and print the news articles based on the newspaper template indicated in personal-news-profile **19**. A description of how the Web news retrieval system of the invention performs this function is described next.

Retrieving a Document Using a Personal-News-Profile

FIG. **6** is a representational block diagram of the manner by which the invention retrieves articles from the Web according to personal-news-profile **19**. (FIG. **6** also shows the manner by which the retrieved articles are flattened into a linear document and formatted. These functions are discussed in greater detail in the next section of this application.)

As shown in FIG. **6**, Web printer **17** is responsible for retrieving news articles. Web printer **17** is an end-user application that communicates with personal-news-profile(s) **19**, site profile **20**, Web reader **34**, and output interface **40** in order to perform this function.

Web printer **17** looks at personal-news-profile **19** to determine which Web sites to access and which data to retrieve from those sites. Web printer **17** also looks at site profile **20** for general site information. According to the information in personal-news-profile **19** and site profile **20**, Web printer **17** instructs Web reader **34** to connect to the Web via Web server **35** in order to access various Web sites and to retrieve data from those sites. Web reader **34** sends the

retrieved data to Web printer **17**, and Web printer **17** uses the data to build an extracted data tree. As will be discussed in greater detail in the next section of the application, Web printer **17** then flattens the extracted data tree into a linear document and formats the linear document for output via output interface **40**.

As shown in FIG. **6**, Web printer **17** includes four program modules: Web reader interface **50**, site driver **51**, tree manager **41**, and formatter **42**.

Web reader interface **50**, like Web reader interface **37** described above, interfaces Web printer **17** to Web reader **34**.

Site driver **51** accesses site profile **20** and personal-news-profile **19** and provides data stored therein to Web reader **34**. As noted above, Web reader **34** uses that data to access various Web sites and to extract data therefrom. As noted above, this retrieved data is used by Web printer **37** to build an extracted data tree.

Tree manager **41** manages the extracted data tree. In this regard, tree manager **41** keeps track of the organization of the retrieved data in the extracted data tree. This allows Web printer **17** to avoid accessing the same article twice, to avoid unnecessarily re-visiting a Web site, and to avoid getting caught in a cycle (loop) in the organization of a hypermedia document on the Web. Alternatively, tree manager **41** could store the data in blocks (as opposed to directly in a data tree) with reference to a linked list that provides the same functionality as the extracted data tree. Sample code for tree manager **41** is included in Appendix 3B.

Formatter **42** is responsible for flattening the extracted data tree into a linear document and formatting the linear document into a personalized newspaper. Formatter **42** performs these functions in accordance with the print criteria and format information (i.e., newspaper template) indicated in personal-news-profile **19**. Sample code for formatter **42** is included in Appendix 3B.

In more detail, FIG. **7** is a flow diagram describing how Web printer **17** uses Web reader **34** to traverse the Web according to personal-news-profile **19** and to retrieve articles from the Web according to the profile, excluding unwanted data.

The Web printer starts in step **S700**. In step **S701**, Web printer **17** retrieves either a user designated personal-news-profile or a default personal-news-profile stored in disk drive **5** using site driver **51**. In this regard, because computer equipment **1** may be used by more than one user, there may be one or more personal-news-profiles stored on the equipment, one of which will be designated as the default. Upon retrieving the designated personal-news-profile, in step **S702** Web printer **17** determines whether any news data has been previously stored to disk drive **5** (for example, by a previous automatic news delivery) or if news articles should be retrieved using personal-news-profile **19**.

In the case that news data does exist on disk drive **5**, in step **S703** the stored news data is retrieved and flow proceeds to step **S801** of FIG. **8**, discussed in more detail in the next section. On the other hand, if no stored news data exists, Web printer **17** invokes Web reader **34** in step **S704**. Note that this is the same Web reader **34** as discussed above with respect to defining a personal-news-profile.

Upon being invoked, Web reader **34** connects to Web server **35** in step **S705**, which provides a connection to a network, such as the World Wide Web. Web printer **17** then provides Web reader **34** with an address for the first Web site to be visited based on information retrieved from personal-news-profile **19**. Once connected to the desired Web site in step **S706**, Web printer **17** provides Web reader **34** with commands/links for traversing the Web to the next Web page containing information that personal-news-profile **19** indicates should be retrieved. Web reader **34** traverses the Web according to this information in step **S707**.

In step **S708**, Web reader **34** retrieves the desired information and sends it to Web printer **17** according to the rules in personal-news-profile **19**. Thus, data exclusion occurs in this step. The rules in personal-news-profile **19** specify structural and content-based criteria for excluding data from the personalized newspaper. The structural rules limit the retrieved information on the basis of the structure of the Web site accessed by Web reader **34**. The content-based rules limit the retrieved information on the basis of its content. As mentioned above with respect to creating a personal-news-profile, examples of the syntax of the retrieval rules in personal-news-profile **19** are included in Appendix 2.

In addition to rule-based exclusion, media-type exclusion occurs in step **S708**, wherein data of a media type that can not be printed is excluded from the extracted data tree. For example, movie and sound data can be excluded.

Web printer **17** stores the retrieved data in disk drive **5** (or in main memory **14**) in the extracted data tree managed by tree manager **41**. Alternatively, the data could be stored in blocks with reference to a linked list, as discussed earlier. In step **S709**, Web printer **17** returns to step **S707** to complete retrieving all information from Web pages at the Web site. In step **S710**, upon completing a traversal of one Web site, Web printer **17** uses tree manager **41** to compare the sites remaining in personal-news-profile **19** with the site organization information in the extracted data tree to determine if more sites need to be visited. In the case that more Web sites need to be visited, step **S710** returns flow to step **S706** and news articles are retrieved in the same manner as discussed above. On the other hand, if all of the Web sites listed in personal-news-profile **19** have been visited and all of the articles retrieved, flow proceeds to step **S801** in FIG. **8**.

Flattening and Formatting the Retrieved Data

FIG. **8** is a flow diagram showing how the extracted data tree is flattened and formatted. The configuration of the invention is the same as when retrieving data from the Web (shown in FIG. **6**). In fact, the flattening and formatting processes can occur, at least to a limited extent, concurrently with the data retrieval process.

In step **S801** of FIG. **8**, the extracted data tree is flattened. This simply means that the organization of the data is converted from an extracted data tree to a linear document. This step provides the opportunity for excluding more data from the personalized newspaper, for example by only including nodes of the data tree into the flattened document. This exclusion process is controlled by the flattening rules in personal-news-profile **19**.

After the data is flattened into a linear document, the data is formatted in step **S802** according to the template indicated in personal-news-profile **19**. The definition of this template, which is either a pre-defined template or a custom template, was discussed earlier. Finally, in step **S803**, the formatted and fully personalized newspaper is sent to output interface **40**. This interface could be printer interface **10** to printer **7**, display interface **11** to display **2**, or even modem/fax interface **6**.

Second Embodiment: The HTML Formatter

The second embodiment of the invention is a system for processing a hypermedia document. The system accesses the

hypermedia document, extracts addresses from the hypermedia document, and stores the addresses extracted from the hypermedia document in a container. The system activates a processing function to process data stored at the addresses stored in the container, downloads the data stored at the addresses stored in the container into a memory, and extracts predetermined data from downloaded data in accordance with predetermined configuration information. The predetermined data is then formatted in accordance with predefined formatting settings to generate a formatted document, and the formatted document is processed in accordance with the processing function.

The second embodiment of the invention is depicted as HTML formatter **18**, noted in FIG. **2**. An example of HTML formatter **18** is WebFormatter, manufactured by Canon Information Systems, Inc. The second embodiment will be described with respect to WebFormatter. It should be noted, however, that HTML formatter **18** is not limited to the WebFormatter embodiment, and that various alternative embodiments within the spirit and scope of the following description are possible.

WebFormatter is stand-alone utility software that can be used in conjunction with different Web browsers, such as Netscape, Mosaic and Internet Explorer. In short, WebFormatter extracts data from a Web page, strips out extemporaneous data from the extracted data, and reformats the data into a formatted document. The formatted document can then be printed, stored in an RTF (Rich Text Format) file, or edited in any RTF compatible editor, such as MS Word, WordPerfect, Wordpad, etc.

WebFormatter can be activated from a windowing environment, such as Microsoft Windows®. From such a windowing environment, WebFormatter can be activated by double-clicking on a WebFormatter icon (not shown) in a start-up window, selecting WebFormatter from the Windows start menu, dragging a URL (uniform resource locator) icon (not shown) from a Web browser and dropping it into the WebFormatter icon, or by automatically invoking WebFormatter when the Web browser is started.

Unlike the first embodiment of the invention described above, WebFormatter does not use a predefined personal-news-profile to specify criteria for creating a particular type of document from one or more Web pages. Rather, WebFormatter relies upon user-specified criteria to create a particular type of document, such as a newspaper or the like, from one or more Web pages. These criteria are input interactively by a user via a graphical user interface.

As described in more detail below, WebFormatter operates in two modes—a minimized mode and a fully-functional mode. In the minimized mode, WebFormatter's graphical user interface is essentially a floating print button, which is displayed concurrently with displayed Web pages. By virtue of this feature, as a user explores the Web, the user can process, format, and print out Web pages by merely clicking on the floating print button.

In its fully-functional mode, WebFormatter's graphical user interface provides spaces for a user to enter a URL address of a Web page to be processed, enter a personal title for the document, select a format for the document, preview a formatted first page of the document, and either print the document, save the document as an RTF file, or view/edit the document using an RTF editor. The graphical user interface for the fully-functional mode will be described first, since it is from that interface that the user can enter the minimized mode.

FIG. **9A** shows graphical user interface **43** for WebFormatter's fully-functional mode. Graphical user interface **43** is displayed on display **2** upon first activation of WebFormatter. As with any interactive windowing software application, a user interacts with graphical user interface **43** by means of mouse **4** (by pointing and clicking) and keyboard **3**.

As shown in FIG. **9A**, graphical user interface **43** includes fields **44** and **46** to **49**, through which a user can specify the URL address of a document to be formatted and the format of that document. Beginning with URL field **44**, a user enters the URL address (e.g., http://www.cis.canon.com/tis/tis_home.htm) of a Web page to be processed by WebFormatter. There are several different ways for the user to enter the URL address. The user can (1) type the address directly into URL field **44**, (2) copy the URL address in the Web browser and paste the URL address into URL field **44**, (3) drag the URL address from the Web browser onto graphical user interface **43** or onto the WebFormatter icon, or (4) click on Current URL button **54**.

With regard to Current URL button **54**, if a user clicks on Current URL button **54**, WebFormatter locates the active Web browser and queries the Web browser for the address of the current Web page. Thereafter, the Web browser provides the address of the current Web page to WebFormatter, which places the address in URL address field **44**. If URL button **54** is activated and no Web browser is currently running, WebFormatter displays dialog box **56**, shown in FIG. **9A**.

As shown, dialog box **56** includes Cancel button **57** and Launch Browser button **59**. Cancel button **57** cancels a user's request to input a URL address into URL address field **44** via Current URL button **54**. Launch Browser button **59**, on the other hand, launches a Web browser specified in WebFormatter. As noted below, WebFormatter is configured beforehand with predefined information including a Web browser to be used with WebFormatter. Configuration of WebFormatter will be described in more detail below.

In alternative embodiments of WebFormatter, a filename can also be entered into URL address field **44**. For example, in these alternative embodiments, if a user wishes to format a hyper-linked manual into a book-like format, the user enters the filename into URL address field **44**. Thereafter, WebFormatter proceeds through the file in the same manner as through specified Web pages in order to reformat the hyper-linked manual as desired.

Returning to graphical user interface **43**, title field **46** enables a user to enter a personalized title for a formatted document. The title may be typed directly or pasted into title field **46**.

Formatting fields **47** to **49** define the format of a document to be output by WebFormatter. Options for the different formatting fields can be accessed by clicking on a scroll bar, such as scroll bar **55**, of a respective formatting field. Each of these fields is described in detail below.

Styles field **47** provides four options for formatting an output document. These styles relate to characteristics of an output document such as size of headers, margins, etc. The style options include Contemporary, Formal, Fun and Professional. The invention, of course, is not limited to these four style options, and other styles can be added as desired.

Columns field **48** defines the number of columns in a formatted output document. Two columns options are available—Single and Multiple; however, the invention is not limited to these two options. The Single option, as might be expected, formats the document into a single column. The Multiple option, on the other hand, formats the document into a predetermined number of columns. In preferred embodiments of the invention, the multiple option is set to two columns; however, any number can be set.

Spacing field **49** defines the spacing between lines in a formatted output document. Three options are provided in WebFormatter, but other options can be added as desired. These three options are Condensed, Normal and Easy To Read, with Condensed being the least amount of spacing between lines and Easy To Read being the most amount of spacing between lines.

Graphical user interface **43** is also provided with Preview button **60**. By clicking on Preview button **60**, a user can preview a first page of a formatted document in viewing area **61**. An example of a previewed formatted document is shown in FIG. **9**A.

As shown in FIGS. **9**A and **9**B, WebFormatter also includes Options button **61**. Options button **61** provides a user with additional formatting options which are used by WebFormatter to create a formatted document. A user can activate Options button **61** by clicking thereon. This causes Options dialog box **62**, shown in FIG. **9**B, to appear on display **2**.

As shown in FIG. **9**B, options dialog box **62** includes General options **64**, Container options **66** and Strip Meta Info options **67**. General options **64** includes "Text only" listbox **72**, "Index of links in the page" listbox **73**, and "No floating pictures" listbox **74**. These options are indicated as being selected by a check mark or the like in a respective listbox. As will become clear from their descriptions, more than one of the options in General options **64** can be selected at the same time.

"Text only" listbox **72** instructs WebFormatter to strip all graphics in a Web page and print only text therein. "Index of links in the page" listbox **73** instructs WebFormatter to add a list of all URLs present in a Web page or pages to the end of a formatted document. Preferably, the list of URLs is printed as superscript, and anchor positions of the URLs in the list are marked in bold. "No floating pictures" listbox **74** instructs WebFormatter to print all images in the document in a particular area of the formatted document. In some cases, therefore, when this option is selected, WebFormatter shrinks images, as needed, so that images fit into a particular area.

Strip Meta Info options **67** provides engineering options which facilitate stripping of unnecessary information from a Web page being processed by WebFormatter. The options include (1) "None", which instructs WebFormatter to strip nothing from the Web page, (2) "Till the first horizontal rule", which instructs WebFormatter to strip all links and images until and up to predefined first and second horizontal formatting rules (e.g., up until a horizontal line across a page), and (3) "Till the first text", which instructs WebFormatter to strip all links and images up to first and last occurrences of text in the Web page. Only one of Strip Meta Info options **67** can be selected at a time. Selection thereof is indicated by a dot in a bullet located next to an option, as shown in FIG. **9**B.

Container options **66** provides options for processing documents, addresses for which are stored in container **76** shown in FIG. **9**B. Prior to describing Container options **66**, a description of container **76** will be provided.

As noted, container **76** stores URL addresses of selected documents. Document addresses which are input to field **44** are added to container **76**. The order in which URLs are input into container **76** denotes the order in which data in the URLs is processed by WebFormatter. As shown in FIG. **9**B, once container **76** becomes full, its icon changes to that shown by reference numeral **77**.

When a user clicks on the icon for container **76**, menu **77** is displayed. Menu **77** provides five options; i.e., Open **79**, Empty **80**, Print **81**, Edit **82** and Save **84**. These options are highlighted when activated, and are described in detail below.

Open **79**, when activated, displays Container Contents screen **87** shown in FIG. **9**B. Container Contents screen **87** shows the URL addresses stored in container **76**. Container contents screen **87** provides four buttons; i.e., Add current URL button **88** which adds the current URL to container **76**, Delete button **89** which permits a user to highlight and delete a URL in container **76**, Empty button **90** which permits a user to empty container **76**, and Done button **91** which permits a user to close Container Contents screen **87**. It is noted that a user can also empty the contents of container **76** by clicking on Empty **80** of menu **77**.

In addition, the user can rearrange the order of URLs stored in container **76** by dragging and dropping different URLs at different locations therein. As noted above, since the URLs are processed in the order that they appear in container **76**, this feature permits a user to rearrange the processing order of the URLs in container **76** interactively.

Print **81**, Edit **82** and Save **84**, when activated, cause WebFormatter to download all data at Web pages defined by the URLs stored in container **76**, format them as specified by the user, create RTF file(s) storing the formatted Web pages, and do the selected action, i.e., save, edit or print the RTF file(s). This process is described in greater detail below.

Referring back to Options dialog box **62**, Container options **66** include "Print table of contents" listbox **92** and "Empty after processing" listbox **94**. As shown, a check mark appears in a listbox to indicate that the listbox has been selected. In this regard, more than one listbox can be selected at a time. "Print table of contents" listbox **92**, when selected, instructs WebFormatter to print titles of all URLs in container **76** as a table of contents in a formatted output document. "Empty after processing" listbox **94**, when activated, instructs WebFormatter automatically to empty container **76** after printing, editing or saving a document, without waiting for a user to do so.

Also shown as part of Container options **62** are Select RTF Editor button **69**, Cancel button **70** and OK button **71**. By clicking on Select RTF Editor button **69**, a user can select an RTF file editor, examples of which are noted above. This can be done, for example, by displaying another dialog box listing predefined RTF editors (not shown) and selecting one of the predefined RTF editors. Cancel button **70** cancels Container options **62** and OK button **71** confirms selected options in Container options **62** and then closes its dialog box.

As shown in FIG. **9**B, graphical user interface **43** also includes print icon **96**, edit icon **97**, save icon **99**, help button **100**, done button **101** and minimizing icon **102**. A user may select any of these features by clicking thereon using a mouse.

Print icon **96** opens a print dialog box (not shown), which allows a user to print any number of copies of Web pages formatted by WebFormatter. Edit icon **97** opens an RTF file storing formatted Web page(s) for editing by a predetermined RTF editor. Save icon **99** opens a save dialog box (not shown), which allows the user to name and save a formatted Web page as an RTF file. Help button **100** provides help messages for operating WebFormatter, and Done button **101** exits from WebFormatter. Minimizing icon **102** activates the minimizing mode of Webformatter which was mentioned above and which is described in greater detail below.

FIG. **9**C shows menus provided by WebFormatter during its operation. These menus include file menu **103**, edit menu

104 and window menu 106. File menu 103 provides "Save", "Edit" and "Print" options, the functions of which are identical to those of Save icon 99, Edit icon 97 and Print icon 96, respectively. An "Exit" option is also provided to exit from File menu 103. Finally, File menu 103 provides "Open HTML file" option 107. This option provides a user with the capability to open a local HTML file; i.e., a hypermedia file resident on the user's computer such as a file saved from NetScape, or URL files created by dragging and dropping a URL onto the windows desktop. "open HTML file" option 107 also provides hooks needed to open files created by other Web-file-processing products so that those files can be formatted as RTF files and printed, saved and/or edited using WebFormatter.

Edit menu 104 provides "Paste URL" option 109. "Paste URL" option 109 pastes the contents of a paste buffer, such as a URL address copied from a Web page, into URL field 44, as described above.

Window menu 106 provides a "Help Topics" option which provides a user with information regarding the use, maintenance and background of WebFormatter, and an "About WebFormatter" option which provides a user with a dialog box (not shown) containing WebFormatter's version number and copyright notice(s). Window menu 106 also includes "Preferences" option 110. "Preferences" option 110 opens preferences dialog box 112, shown in FIG. 9D.

Preferences dialog box 112 is used to configure and re-configure WebFormatter. As shown in FIG. 9D, preferences dialog box 112 includes Minimize view options 113, General options 114 and WWW Browser to use options 115. Minimize view options 113 can be set to configure WebFormatter's graphical user interface in the minimized mode. Two sets of options are provided. The first set include "Print", "Edit" and "Save". These options correspond to print icon 96, edit icon 97 and save icon 99, shown in FIG. 9B. When a check mark appears in a listbox next to one of these options, the icon for that option is displayed in the minimized mode, e.g., the print icon, the edit icon and/or the save icon. More than one option can be selected at once. In this regard, FIG. 9E shows graphical user interface 116, which is a representative example of a graphical user interface for WebFormatter when WebFormatter is in the minimized mode.

Referring back to FIG. 9D, Minimize view options 113 also include "Row" and "Stack" options. These options can be set to display WebFormatter's graphical user interface in the minimized mode horizontally by selecting "Row" or vertically by selecting "Stack". Only one of these options can be selected at a time. As an example of the foregoing, graphical user interface 116 corresponds to a row of icons.

WWW Browser to use options 115 determine which World Wide Web browser is to be used with WebFormatter. As shown, preferably NetScape, Internet Explorer and Mosaic are provided as browser options; however, other browser options can also be provided. As might be expected, only one of these options can be selected at a time. The default browser option is NetScape Navigator.

General options 114 include "Auto-start with browser" option 117, "open in minimized view" option 118, "Warn before printing more than _pages" option 119, and "Warn before saving more than _MBs" option 120. "Auto-start with browser" option 117 sets WebFormatter to be invoked automatically when a Web browser is activated. If this option is not selected (which is the default), WebFormatter is opened by double clicking on a WebFormatter icon in the windowing environment, selecting WebFormatter from the Windows start menu, or dragging and dropping a URL from the Web browser into the WebFormatter icon, as described in more detail above. "Open in minimized view" option 118, when selected, opens WebFormatter in minimized mode. The default, however, is the fully-functional mode. "Warn before printing more than _pages" option 119, and "Warn before saving more than _MBs" option 120 allow a user to control the number of pages saved of a formatted document and the amount of memory space used by those pages, respectively. The default for both of these options is for no warning to be given. As is evident, more than one of the general options can be selected at the same time.

Preferences dialog box 112 also includes cancel button 121 which cancels a user's selected preferences and OK button 122 which confirms a user's selected preferences.

As explained above, WebFormatter can be configured to enter directly into the minimized mode via Preferences dialog box 112, or a user can enter the minimized mode via minimizing icon 102 shown in FIG. 9B. As also noted above, FIG. 9E shows an example of graphical user interface 116 for WebFormatter in the minimized mode. Graphical user interface 116 is displayed as a floating interface while a user is exploring the Web. Thus, as a user views a Web page, the user also views graphical user interface 116. By clicking on an appropriate icon on graphical user interface 116 (which, in FIG. 9E, includes icons identical in both structure and function to those shown in graphical user interface 43), the user can capture the current Web page, process and format the Web page into an RTF file, and save, edit and/or print the RTF file. Alternatively, the user can drag a URL from the Web browser and drop it into one of the icons.

A user can reconfigure WebFormatter in the minimizing mode by double clicking a right mouse button. This action causes a preferences dialog box to appear on display 2 which is identical to preferences dialog box 112. Thereafter, the user can alter the configuration of WebFormatter as desired. Should a user wish to enter the fully-functional mode from the minimizing mode, the user need merely click on maximizing icon 117 shown in FIG. 9E.

FIG. 10 is a flow diagram describing the operation of WebFormatter. WebFormatter is activated in step S1000. As described above, this can be done by double-clicking on a WebFormatter icon in a windowing environment. Depending upon how WebFormatter has been configured, i.e., in the fully-functional mode or the minimizing mode, either a graphical user interface similar to that of graphical user interface 43 or one similar to that of graphical user interface 116 is displayed in step S1000. For the sake of completeness, the following assumes that a graphical user interface similar to that of graphical user interface 43 is displayed in step S1000, since the default mode of WebFormatter is the fully-functional mode.

Next, in step S1001, WebFormatter is configured, as described above via preferences dialog box 112 and options dialog box 62. This step is not necessary unless a user wishes to change WebFormatter's previously set configuration. In step S1002, document format data is input in fields 44 and 46 to 49 described above. More specifically, the user inputs a URL (or filename in alternative embodiments) into URL field 44. As described below, WebFormatter uses this information to process Web pages stored at the URL to create an RTF file based on the configuration of WebFormatter and the data input in fields 46 to 49.

In step S1003, a Web reader similar to that of Web reader 34 described above is executed. The Web reader connects to a network, such as the World Wide Web, in step S1004.

Next in step S1005, it is determined whether a URL or a filename has been entered. As described above, in preferred embodiments of WebFormatter, only a URL may be entered. However, since alternative embodiments of WebFormatter may permit entry of a filename, a description of processing a file other than one at a URL address will be provided.

If a URL has been entered in field **44**, processing proceeds to step S1006. In step S1006, the Web reader accesses the hypermedia document (e.g., a homepage) specified by the URL address. In step S1007, WebFormatter instructs the Web reader to traverse the hypermedia document. Thereafter, WebFormatter selects URL address(es) from the Web and stores the addresses in container **76**. Once all desired addresses have been selected and a processing function, such as print, has been activated, WebFormatter downloads data stored at the addresses in container **76** into memory **5**. WebFormatter then extracts predetermined data from the downloaded data based on the configuration information set in Optional dialog box **62**, and stores the extracted data in memory **5**. Thus, for example, if "Text Only" option **72** in Options Window **62** is on, only text is extracted from the downloaded data. Processing then proceeds to step S1011.

On the other hand, if, in step S1005, a filename for an HTML source file is entered, WebFormatter instructs the Web reader to access a first site in the file. In steps S1008 and S1009, the site is traversed and data is extracted and stored in the same manner as in step S1007, described above. Then, in step S1010, WebFormatter determines if more sites are listed in the HTML source file. If more sites are listed in the file, flow returns to step S1008, and the next site is accessed. If no more sites are present, processing proceeds to step S1011,

In step S1011, WebFormatter processes the extracted data in accordance with the previously set format information. For example, if Columns field **48** is set to multiple, the extracted data will be formatted into a document having multiple columns. The above processing is initiated by activating one of Print icon **96**, Edit icon **97** or Save icon **99**, and is similar to the processing described above in the first embodiment, e.g., flattening the document and formatting the document based on the formatting information. Accordingly, a detailed description thereof is omitted for the sake of brevity.

Once the documents whose URLs are stored in the container have been downloaded, formatted according to the preset formats and configurations, and converted into RTF file(s) in step S1011, in step S1012, the RTF file(s) are output. Alternatively, the RTF files(s) can be edited or saved, depending upon which icon on the graphical user interface has been activated.

The invention has been described with respect to particular illustrative embodiments. It is to be understood that the invention is not limited to the above described embodiments and modifications thereto, and that various changes and modifications may be made by those of ordinary skill in the art without departing from the spirit and scope of the appended claims.

APPENDIX 1

SAMPLE USER PROFILE

```
The User Profile is implemented in windows.ini file format.
[Defaults]
Count = 4
Title = My Daily Paper
[1]

Heading = News In Brief
Site = 1
Section = Front Page
MaxLevels = 5
Maxpages = 10
MaxKBytes = 2000
Date = today
Print = level 0
Template = 1
[2]

Heading = Sports In Brief
Site = 2
Section = Sports
Max Levels = 0
MaxPages = 10
MaxKbytes = 200
KeywordFilter = "Football" AND "49ers"
Date = today
Print = level 0
Template = 1
[3]

Heading = Money Matters
Site = 1
Section = Business
MaxLevels = 1
MaxPages = 100
MaxKBytes = 20000
KeywordFilter = "Computer" OR "hardware" OR
"Software"
Date = today
Print = all
Template = 2
[4]

Heading = Sri Lanka
Site = 3
Section = HotNews
MaxLevels = 1
MaxPages = 100
MaxKBytes = 20000
Date = today
Print = leaves
Template = 2
```

APPENDIX 1-continued

SAMPLE SITE PROFILES

```
#Legend:
#%W-day of the week
#%s-section part of URL
[Defaults]
Count = 3
[1]

Title = San Jose Mercury News
Username = mwickram
Password = cannon
StartData = StartHeadlines
EndData = EndHeadlines
Home Page = http://www.sjmercury.com/
SectionURL = http://www.sjmercury.com/%S.htm
SectionCount = 9
Section 1 = Front Page
Section 2 = International
Section 3 = National
Section 4 = Local & State
Section 5 = Editorials Commentary
Section 6 = Business
Section 7 = Sports
Section 8 = Living
Section 9 = Entertainment
[1. Sections]

Front Page = front
International = intl
```

APPENDIX 1-continued

National = natl
Local & State = loc
Editorials & Commentary = edit
Business = biz
Sports = spts
Living = liv
entertainment = ent
[2]

Title = The San Francisco Chronicle
Home Page = http://www.sfgate.com/chronicle/
SectionURL = "http:www.sfgate.com/cig-bin/chronicle/article-list.cgi?%/S:/chronicle/today"
Section Count = 5
Section 1 = News
Section 2 = Business
Section 3 = Sports
Section 4 = Editorial
Section 5 = Datebook
[2. Sections]

News = News:MN
Business = Business:BU
Sports = sports:SP
Editorial = Editorial:ED
Datebook = Datebook:DD
[3]

Title = The Day News
Hoxne page = http://www.landa.net/lakehouse/anclWeb/dailynew/
SectionURL = "http://www.lanka.net/lakehouse/anclWeb/dailynew/%W/WS.html"
SectionCount = 12
Section 1 = Business
Section 2 = Editorial
Section 3 = Features
Section 4 = Foreign
Section 5 = Letters
Section 6 = InBrief
Section 7 = HotNews
Section 8 = Probes
Section 9 = Military
Section 10 = Politics
Section 11 = Obituaries
Section 12 = Sports
[3. Sections]

Business = business/intro
Editorial = editorial/final
Features = features/intro
Foreign = foreign/intro
Letters = letters/final
InBrief = inbrief/intro
HotNews = hotnews/intro
Probes = proves/intro
Military = military/intro
Politics = politics/intro
Obituaries = obiturai/intro
Sports = sports/intro

APPENDIX 2

SYNTAX FOR RETRIEVAL, EXTRACTION AND PRINTING CRITERIA

Maximuin levels to search: MaxLevels =<#>
-1: to retrieve all levels
O - n: to retrieve up to n levels
Maximum pages of the document: MaxPages = <#>
n: final document not more than n pages
Maximum size of the document: MaxKBytes = <#>
n: document size not more than n kilo bytes
Exclusion rules:
Date = today|lessthan <#>

APPENDIX 2-continued

SYNTAX FOR RETRIEVAL, EXTRACTION AND PRINTING CRITERIA

today: retrieve only articles posted today
lessthan <#>;n: retrieve only articles no more than n days old
Retrieve = all|nosubdir|nothisdir|thissiteonly
all: allow to fetch pages from other sites
nosubdir: exclude URLs to subdirectories
nothisdir: exclude URLs in this directory
thissiteonly: fetch pages from this site only
Keyword search:
KeywordFilter = <keyword> (AND|OR|NOT) <keyword>: accumulate only pages containing the combination of keywords
KeywordRank = <#>;n: use fuzzy logic to rank pages according to keyword combination in KeywordFilter and keep top n ranked pages
KeywordAuthor = <author>: accumulate only pages authored by author
ExcludeType = ads|nonEnglish
ads: exclude advertisements
nonEnglish: exclude articles that are not in English
Flattening rules: Print = all|leaves|level = <#>
all: include all nodes in the tree in the linear document
leaves: include all leaves in the tree in the linear document
level = <#>;n: include up to nth level of the tree in the linear document
Formatting rules: Template = <#>
n: print according to default or user template number n

APPENDIX 3

DESCRIPTION OF MODULES
Appendix 3A

THE PERSONAL NEWS PROFILE EDITOR MODULE

The Profile Editor manages access to the user profiles and is represented by CProfileMgr class. It also manages loading and saving of the profiles. The services provided by Profile Editor are:

BOOL Newprofile(CString fileName);
Creates a new profile given the file name.
BOOL OpenProfile( );
Opens the default profile.
BOOL OpenProfile(CString fileName);
Opens the named profile.
CProfileEntry* GetFirstEntry( );
Loads and returns the next profile entry.
CProfileEntry* GetNextEntry( );
Loads and returns the next profile entry.
BOOL WriteEntry(CProfileEntry& entry);
Saves a new entry in the profile.

Each profile entry contains an extraction specification and an output specification as represented by CProfileEntry class. The methods provided are:

CURL GetSiteId( );
Returns the site id contained in the profile entry.
CExtractionSpec GetExtractionSpec( );
Returns the extraction specification contained in the profile entry. Extraction specification contains keywords for searching, limits for levels, pages, size in kilo bytes.
COutputSpec GetOutputSpec( );
Returns the output specification contained in the profile entry. Output specification contains formatting instructions and tree traversal rules.

THE Web READER MODULE

CWebPage class abstracts the interface to the Internet browser and is representative of the actual Web page. It will be responsible for fetching a Web page, extracting links or references to other URLs in the Web page, and maintaining the contents of a Web page. The methods provided are:

APPENDIX 3-continued

BOOL Load( );
Fetch the Web page using the URL, username and password.
BOOL Parse( );
Parses the data in the Web page and creates a list of links. Also resolves the relative URLs into absolute URLs.
CURLList* GetLinks( );
Returns the list of links in the Web page.
CPageData* GetData( );
Returns the actual text data contained in the Web page.
void FilterContent( );
Extracts title and other information according to the site data.
CString GetTitle( );
Returns title and other information according to the site data.
CString GetAuthor( );
Returns the author of the Web page.
int GetSize( );
Returns the size of the data in kilo bytes.

CNetwork class will encapsulate OLE functionality and provides communication with the Internet browser.

CString GetUsername( );
Determine the currently set username.
void SetUsername(LPCTSTR);
Set the current username in the CNetwork object.
CString GetPassword( );
Determine the currently set password.
void SetPassword(LPCTSTR);
Set the current password in the CNetwork object.
void Close( );
Disconnect any active connection and reset the CNetwork object.
short Read (BSTR*pBuffer, shortiAmount);
Read data retrieved by the Browser.
long GetStatus( );
Query the status of the current load.
BOOL Open(LPCTSTR pURL, shortiMethod, LPCTSTR pPostData, long IPostDataSice, LPCTSTR pPostHeaders);
Initiates the retrieval of a URL from the network.
CString GetErrorMessage( );
Provide the caller with internally generated error messages.
short GetServerStatus( );
Determine the error status reported by the server.
long GetContentType( );
Return the content length (total amount of bytes) of the current load.
CString GetContentEncoding( );
Return the MIME encoding of the current load.
CString GetExpires( );
Return when the data retrieved by this load is no longer considered valid.
CString Resolve(LPCTSTR pBase, LPCTSTR pRelative);
Generate an absolute (fully qualified) URL.
BOOL IsFinished( );
Determine if a load is complete.
short BytesReady( );
Inform the caller of the number of bytes prepared to be read.

THE SITE DRIVER MODULE

The Site Driver will provide the site information to the Web Reader. The Site Driver is functionally similar to the Profile Editor and is represented by CSiteDriver class. Services provided are:

BOOL NewProfile(CString fileName);
Creates a new profile given the file name.
BOOL OpenProfile( );
Opens the default profile.
BOOL OPenProfile(CString fileName);
Opens the named profile.
CSiteProfile* GetFirstSite( );
Loads and returns the first site entry.
CSiteProfile* GetNextSite( );
Loads and returns the next site entry.
BOOL WriteEntry(CSiteProfile& entry);
Saves a new entry in the profile.
int NumberOfSites( );
Returns the number of sites specified in the profile.

An entry in the site profile will contain information about the base URL of the site, title of the news source, information about how to access the site, and various other information such as section data etc. and will be represented by CSiteEntry class. Methods provided are:

CString GetURL( );
Returns the base URL of the site.
CString GetUsername( );
Returns the username for the site.
CString GetPassword( );
Returns the password for the site.
CString GetTitle( );
Returns the password for the site.
CString GetTitle( );
Returns the title of the news source.
int SectionCount( );
Returns

Appendix 3B

TREE MANAGER MODULE

Tree Manager will maintain the most central data structure in this program, which is a tree of Web page nodes and is represented by the CPageTree. CPageTree will traverse the WWW to retrieve the necessary Web pages according to the extraction specificatioin and builds the tree. The methods provided are:

CPageTreeNode* GetRoot( );
Returns the root node of the tree.
BOOL Build(CURL URL, CExtractionSpec& spec);
Builds the tree according to the personal news profile extraction specification.

Each node in the page tree is represented by a CPageTreeNode. Methods provided are:

BOOL AddChild(CWebPage* page);
Adds a child node with Web page data.
CWebPage* GetPage( );
Returns the Web page contained in the node.
int NumberOfChildren( );
Returns the number of children belonging to the node.
BOOL IsLeaf( );
Returns TRUE if a leaf node, i.e., no children.

To traverse the Web page tree, a CTreeIterator class is defined with different traversal methods. Methods provided are:

void Reset( );
Cancels the current transferral, and initializes state data.
CPage TreeNode* GetNextNode( );
Returns the next node in the tree in a depth first search.
CPage TreeNode* GetNextSibling( );
Returns the next node in the tree in a breadth first search.
CPage TreeNode* GetNextLeaf( );
Returns the next leaf in the tree in a depth first search.

THE FORMATTER MODULE

Input to this module will be the Web page tree created by the Tree Manager and the output specification contained in the user profile. Formatter will traverse the tree according to the rules specified in the output specification and the final document will be formatted using the formatting instructions in the output specification and the formatting contained in the Web pages such as headings, paragraphs and lists etc. The output document will be in Rich Text Format (RTF) and will be accessible by many applications. RTF is a advanced formatting language for text, providing document, section and paragraph formatting, style sheets, headers and footers, and with support for Uniconde. Image formats supported are DIB, DDB, WMF, OS/2 metafiles. There is no support for Web images which are of the GIF format. A third party library will need to be purchased in order to do the conversion of the GIF to DIB format or one can be developed in-house.

The prototype creates a HTML file as the output.

The formatter is represented by the CFormatter class. The methods provided are:

BOOL OpenHTMLFile(CString fileName);
Opens the named HTML file for output.
void CloseHTMLFile( );
Closes and saves the HTML file.
BOOL PrintHTML(CPageTree& root, COutputSpec& format);
Given the root and the output specification, traverses the tree and prints the contents in the Web pages in HTML format.
BOOL OpenRTFFile(CString fileName);
Opens the named RTF file for output.

APPENDIX 3-continued

```
void CloseRTFFile( );
        Closes and saves the RTF file.
BOOL PrintRTF(CPageTree& root, COutputSpec& format);
        Given the root and the output specification, traverses the
        tree and prints the contents in the Web pages in RTF format.
BOOL Print(CPageTree& root, COutputSpec& format);
        Given the root and the output specification at, traverses the
        tree and prints the contents in the Web pages to the default
        printer.
```

What is claimed is:

1. An automated method for formatting data into a personalized newspaper from at least one hypermedia document, comprising the steps of:

an accessing step to access the at least one hypermedia document;

a traversing step to traverse selectively links in the hypermedia document;

a retrieving step to retrieve data from the hypermedia document and/or traversed links into an extracted data tree, wherein the data is retrieved based on a structure of the hypermedia document and/or links in the hypermedia document;

a flattening step to flatten the extracted data tree into a linear document; and

a formatting step to format the linear document into a formatted personalized newspaper consisting of text and/or images, wherein a number of links traversed in the traversing step can be limited to a predefined number of links.

2. The method of claim 1, further comprising the step of printing the formatted document.

3. The method of claim 1, wherein said hypermedia document is located on the World Wide Web.

4. The method of claim 1, wherein said hypermedia document is located on the Internet.

5. The method of claim 1, wherein said hypermedia document is located on an intranet.

6. The method of claim 1, wherein said accessing step, said retrieving step, said flattening step, and said formatting step are performed in accordance with a personal-news-profile.

7. An automated method for retrieving articles from a hypermedia-linked computer network and for formatting the articles into a personalized newspaper, the method comprising the steps of:

retrieving a stored personal-news-profile which comprises address data for a site on the hypermedia-linked computer network, command data for accessing data from the site, and newspaper layout commands;

contacting the site based on address data stored in the personal-news-profile;

traversing selectively links in the site;

downloading articles from the site and/or links in the site based on command data stored in the personal-news-profile;

flattening the articles into a linear document; and

formatting the linear document into the personalized newspaper according to layout commands stored in the personal-news-profile, the personalized newspaper consisting of text and/or images,

wherein a number of links traversed in the traversing step can be limited to a predefined numbers of links based on command data in the personal-news-profile.

8. The method of claim 7, further comprising the step of printing the personalized newspaper.

9. The method of claim 7, wherein said hypermedia-linked computer network is the World Wide Web.

10. The method of claim 7, wherein said hypermedia-linked computer network is on the Internet.

11. The method of claim 7, wherein said hypermedia-linked computer network is on an intranet.

12. The method of claim 7, wherein the command data for accessing data includes data for selecting articles based on a structure of the site.

13. The method of claim 12, wherein the command data for accessing data also includes data for selecting articles based on a content of the articles.

14. Computer executable process steps stored on a computer-readable medium, said steps for accessing World Wide Web sites for retrieving data at the sites and for formatting the data into a personalized newspaper, said steps comprising:

a connecting step to connect to the World Wide Web;

a retrieving step to retrieve user-defined Web site address information, user-defined Web site commands, and user-defined formatting commands;

an activating step to activate a Web reader so as to access a Web site based on the user-defined Web site address information, a traversing step for traversing selectively links in the Web site, and retrieving data from within the Web site and/or links based on the user-defined Web site commands;

a downloading step to download the retrieved Web site data and/or link data from the accessed Web site into an extracted data tree;

a flattening step to flatten the extracted data tree into a linear document;

a step to repeat the downloading step and the flattening step until all addresses/links in the user-defined Web site address information have been accessed; and

a formatting step to format the stored data into the personalized document based on the user-defined formatting commands, said personalized document consisting of text and/or images,

wherein a number of links traversed in the Web site can be limited to a predefined number of links based on the user-defined Web site commands.

15. The computer executable process steps of claim 14, further comprising a spooling step to spool the personalized document to an output device.

16. The computer executable process steps of claim 15, wherein the output device is a printer.

17. The computer executable process steps of claim 15, further comprising an output step to output the personalized document to a display.

18. The computer executable process steps of claim 14, wherein the user-defined Web site commands include commands for selecting data based on a structure of the Web site.

19. The computer executable process steps of claim 18, wherein the user-defined Web site commands also include commands for selecting data based on a content of the Web site.

20. An apparatus for automatically retrieving news articles from on-line news services on the World Wide Web and formatting the news articles into a personalized newspaper, the apparatus comprising:

first storage means for storing (1) a personal-news-profile which comprises address data and command data for accessing data from a Web site, and (2) newspaper format commands;

retrieval means for retrieving the stored personal-news-profile and accessing data stored therein;

activating means for activating a Web reader to contact a Web site based on address data stored in the personal-news-profile;

traversing means for traversing selectively links in the Web site;

downloading means for downloading news articles from the contacted Web site and/or links based on command data stored in the personal-news-profile;

second storage means for storing the downloaded news articles; and

formatting means for flattening the downloaded news articles into a linear document and for formatting the linear document into the personalized newspaper based on the newspaper format commands stored in the personal-news-profile, said personal newspaper consisting of text and/or images,

wherein a number of links traversed by the traversing means can be limited to a predefined number of links based on command data stored in the personal-news-profile.

21. The apparatus of claim **20**, further comprising spooling means for spooling the personalized newspaper to a printer.

* * * * *

US005649186A

United States Patent [19]

Ferguson

[11] Patent Number: 5,649,186

[45] Date of Patent: Jul. 15, 1997

[54] **SYSTEM AND METHOD FOR A COMPUTER-BASED DYNAMIC INFORMATION CLIPPING SERVICE**

[75] Inventor: **Gregory J. Ferguson**, Hunt Valley, Md.

[73] Assignee: **Silicon Graphics Incorporated**, Mountain View, Calif.

[21] Appl. No.: **511,832**

[22] Filed: **Aug. 7, 1995**

[51] **Int. Cl.**[6] **G06F 17/30**

[52] **U.S. Cl.** **395/610**; 395/601; 395/603

[58] **Field of Search** 395/601, 602, 395/603, 604, 605, 606, 607, 608, 609, 610, 611, 800, 712; 370/60; 364/188, 191, 192

[56] **References Cited**

U.S. PATENT DOCUMENTS

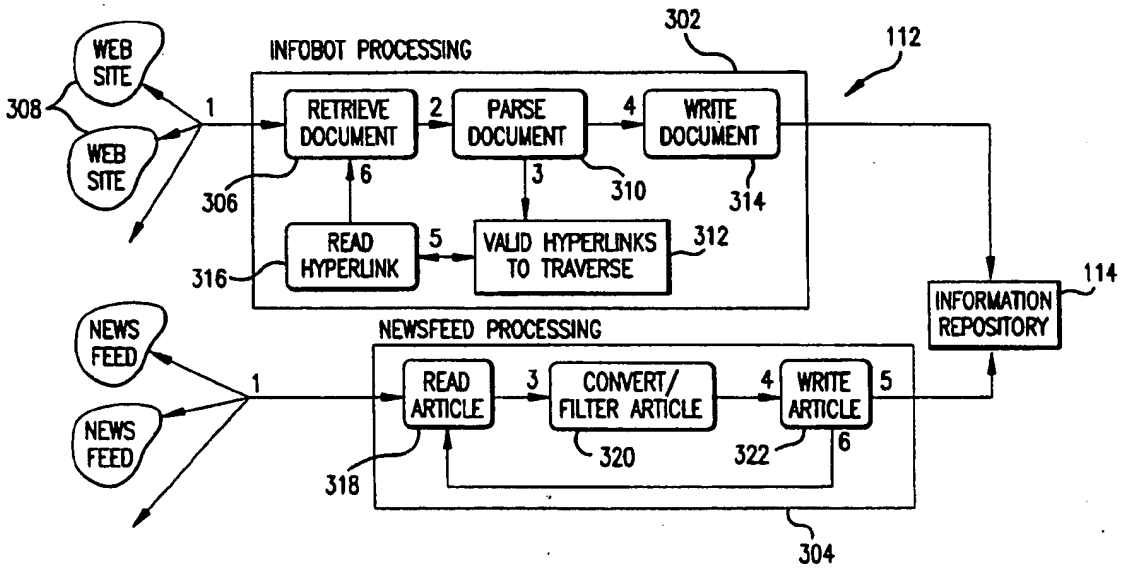
5,019,961	5/1991	Addesso et al.	364/192
5,408,659	4/1995	Cavendish et al.	395/650
5,452,468	9/1995	Peterson	395/800
5,493,568	2/1996	Sampat et al.	370/60

Primary Examiner—Thomas G. Black
Assistant Examiner—Ruay Lian Ho
Attorney, Agent, or Firm—Sterne, Kessler, Goldstein & Fox, P.L.L.C.

[57] **ABSTRACT**

A system and computer-based method providing a dynamic information clipping service. An end-user creates a template of topics of interest via a graphical user interface and the template is transmitted to a central site for processing. At the central site, information relating to a particular base of knowledge is collected, parsed and indexed. The parsed and indexed information is stored in an information repository. The template is processed by parsing and collecting command-strings relating to the topics of interest found within the parsed template. The information repository is searched using the collected command-strings to generate query results, which are then sorted. A Hypertext Mark-up Language (HTML) page is created using the sorted query results. The page is then made available to the end-user for viewing, wherein the page represents a custom network-based newspaper.

14 Claims, 9 Drawing Sheets



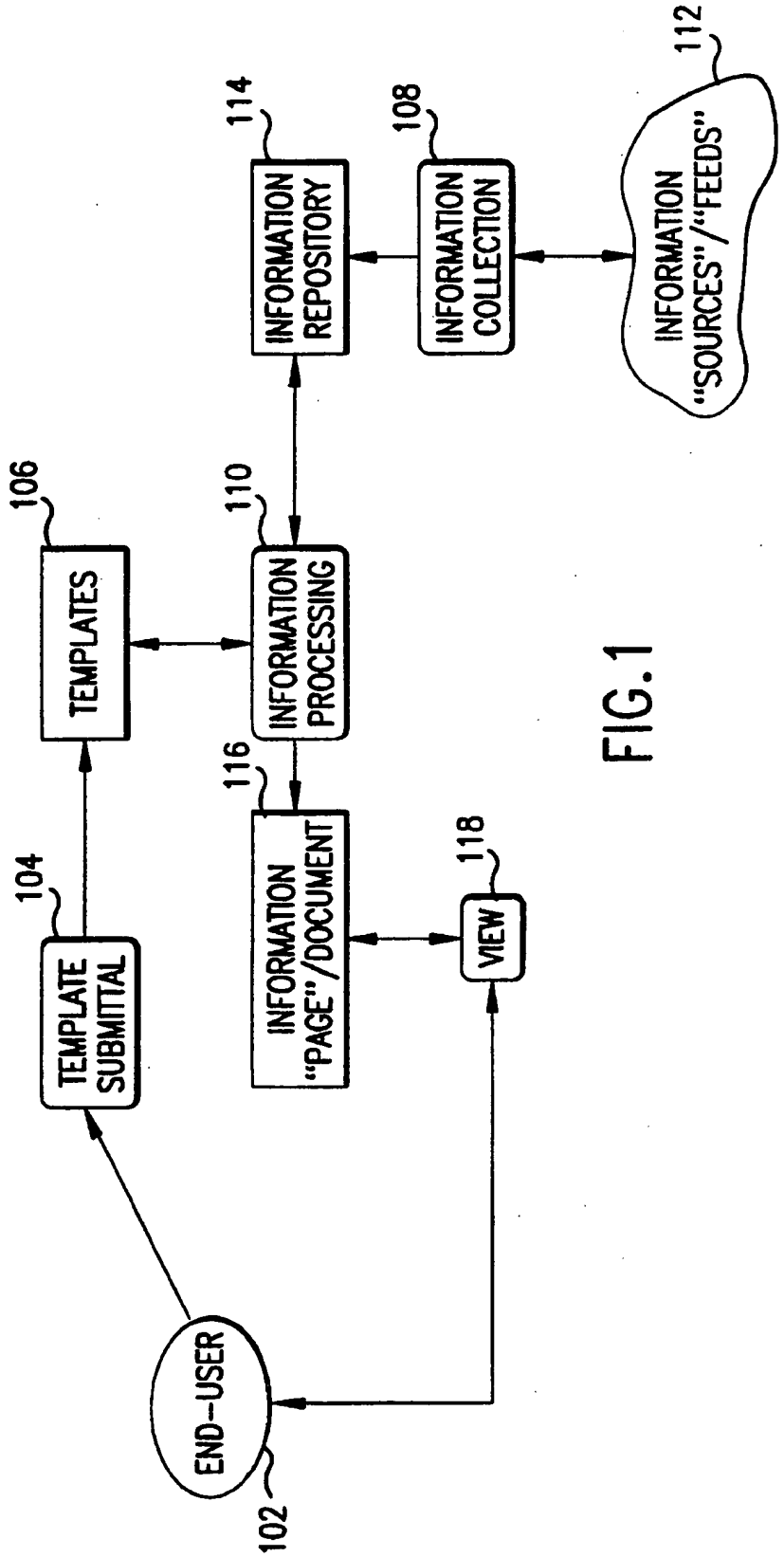

112
INFORMATION "SOURCES"/"FEEDS"
108
INFORMATION COLLECTION
114
INFORMATION REPOSITORY
106
TEMPLATES
110
INFORMATION PROCESSING
116
INFORMATION "PAGE"/DOCUMENT
118
VIEW
104
TEMPLATE SUBMITTAL
END-USER
102

FIG. 1

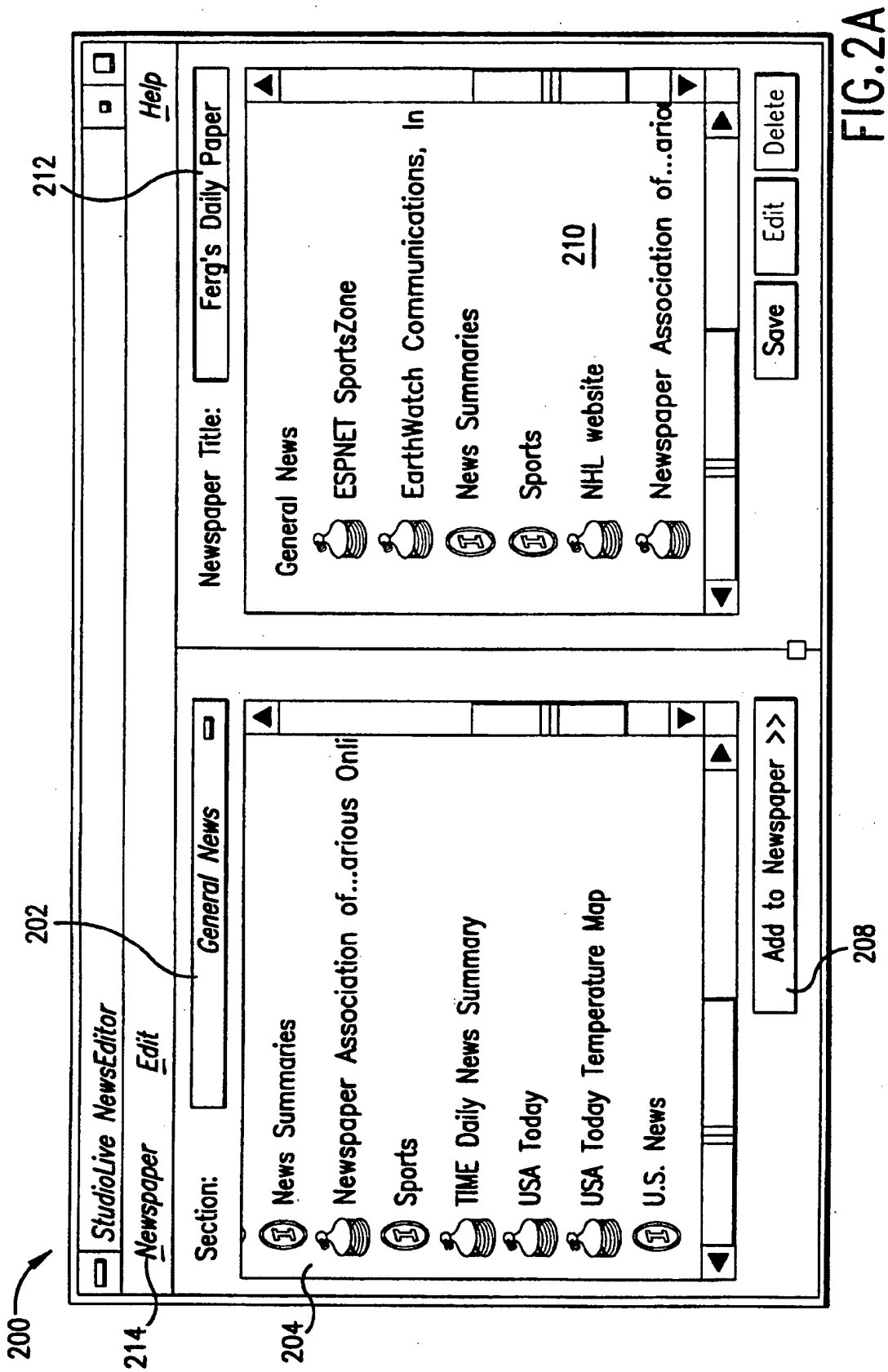

StudioLive NewsEditor
Newspaper Edit Help
Section: General News
News Summaries
Newspaper Association of...arious Onli
Sports
TIME Daily News Summary
USA Today
USA Today Temperature Map
U.S. News
Add to Newspaper >>
Newspaper Title: Ferg's Daily Paper
General News
ESPNET SportsZone
EarthWatch Communications, In
News Summaries
Sports
NHL website
Newspaper Association of...ario
Save
Edit
Delete
200
202
204
208
210
212
214

FIG.2A

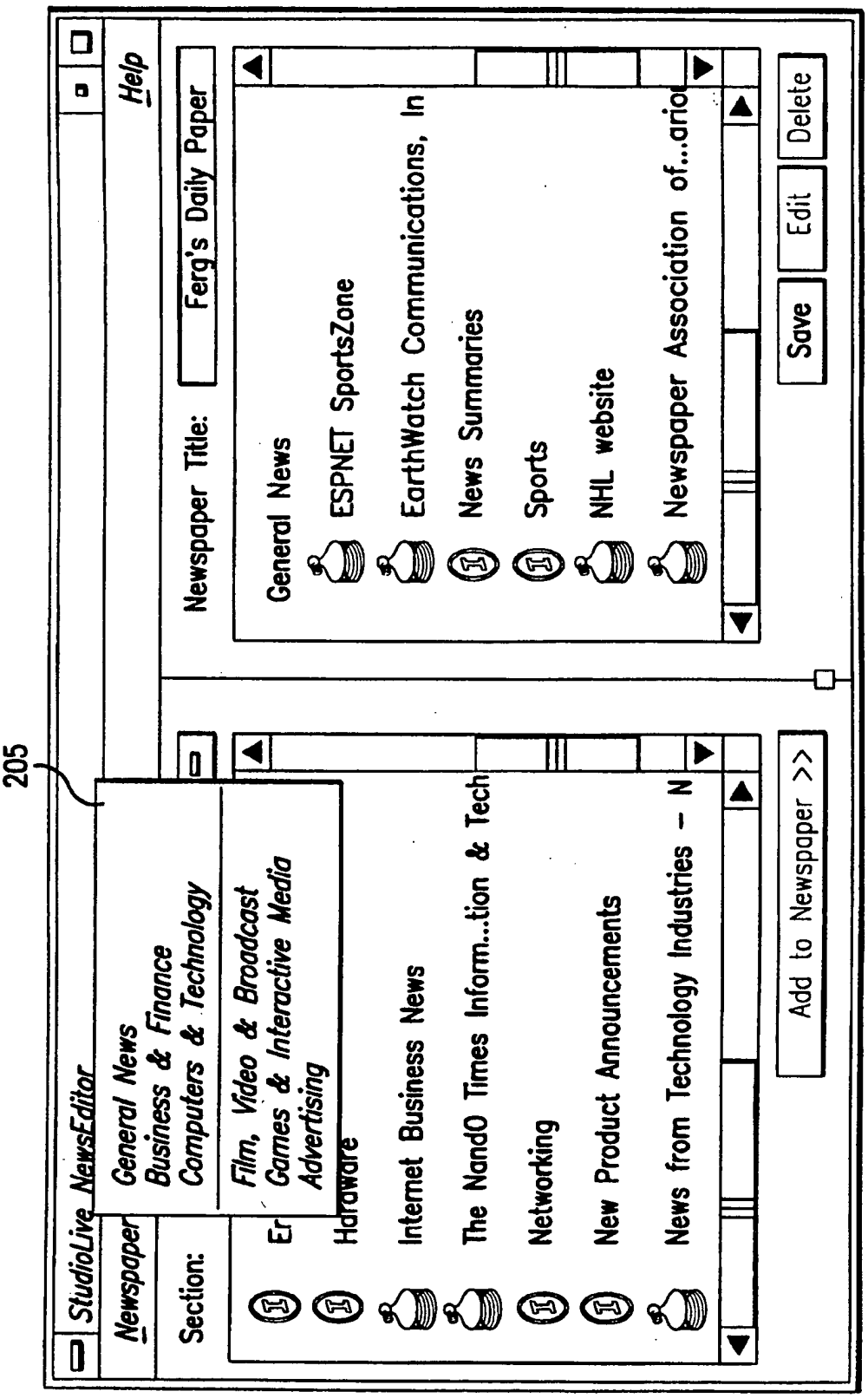

205
StudioLive NewsEditor
Newspaper
Help
Section:
General News
Business & Finance
Computers & Technology
Film, Video & Broadcast
Games & Interactive Media
Advertising
Hardware
Internet Business News
The NandO Times Inform...tion & Tech
Networking
New Product Announcements
News from Technology Industries - N
Add to Newspaper >>
Newspaper Title:
Ferg's Daily Paper
General News
ESPNET SportsZone
EarthWatch Communications, In
News Summaries
Sports
NHL website
Newspaper Association of...ario
Save
Edit
Delete

FIG.2B

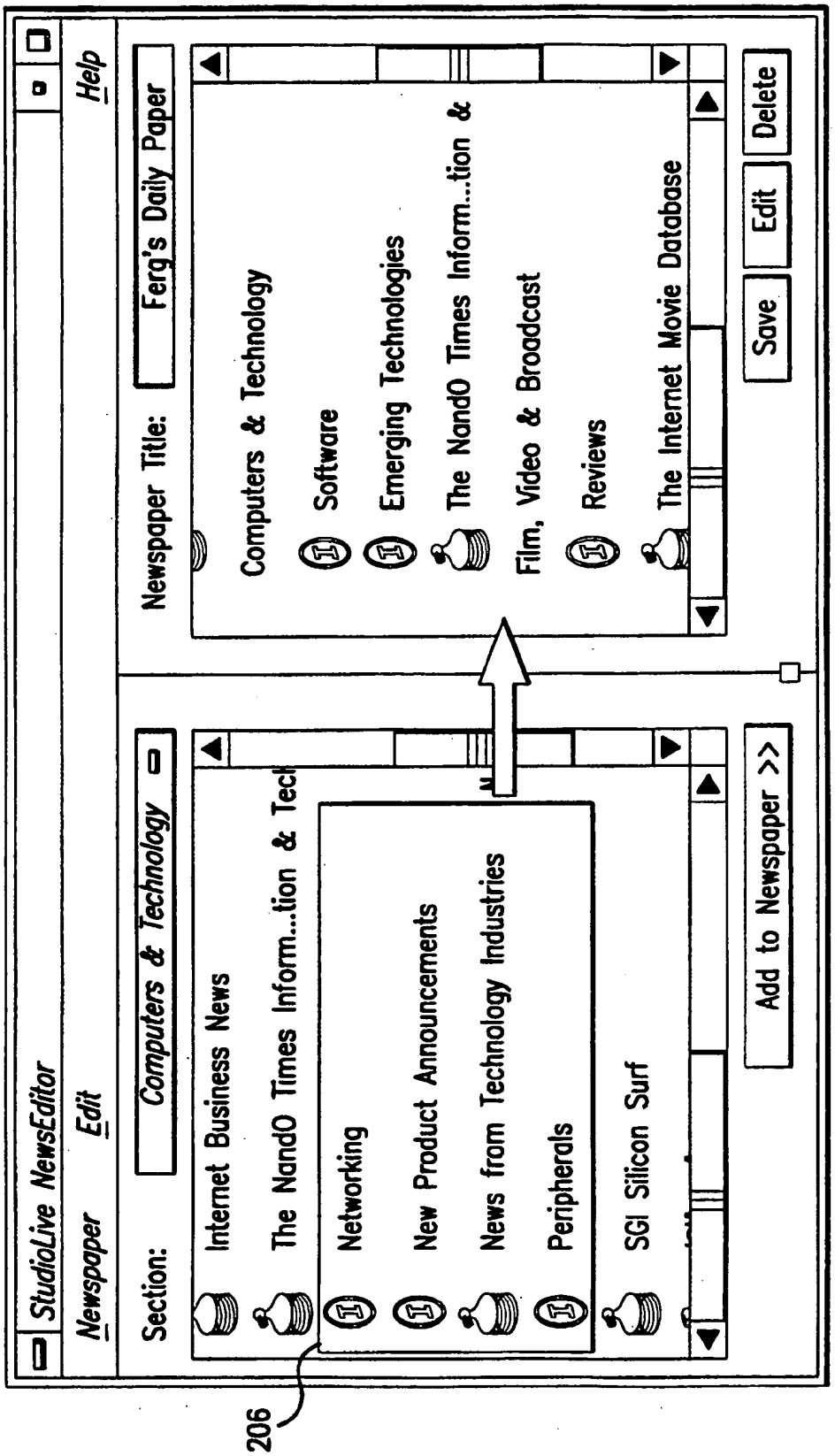

StudioLive NewsEditor
Newspaper Edit Help
Section: Computers & Technology
Internet Business News
The NandO Times Inform...tion & Tech
Networking
New Product Announcements
News from Technology Industries
Peripherals
SGI Silicon Surf
Add to Newspaper >>
Newspaper Title: Ferg's Daily Paper
Computers & Technology
Software
Emerging Technologies
The NandO Times Inform...tion &
Film, Video & Broadcast
Reviews
The Internet Movie Database
Save
Edit
Delete
206

FIG.2C

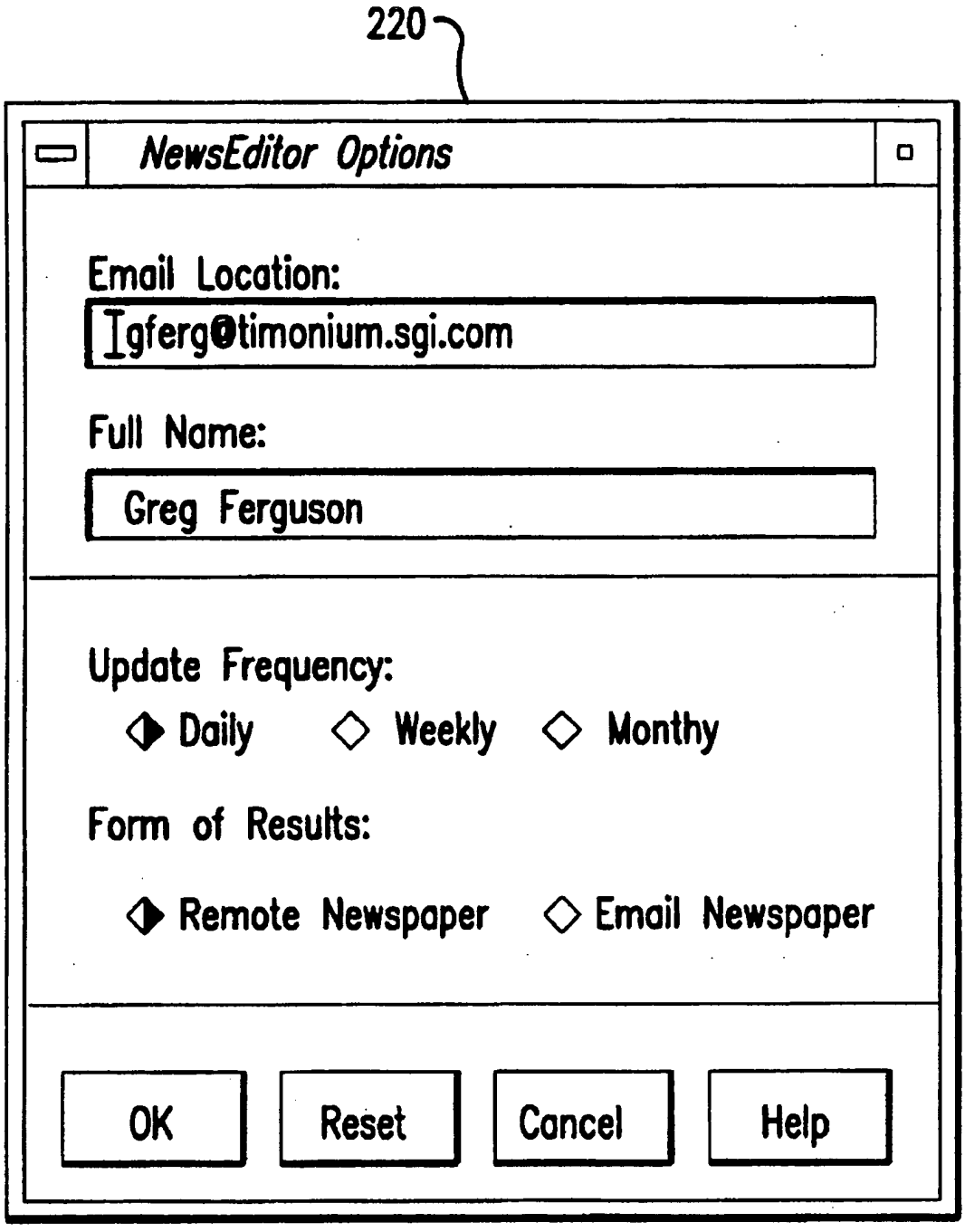

220
NewsEditor Options
Email Location:
Igferg@timonium.sgi.com
Full Name:
Greg Ferguson
Update Frequency:
Daily
Weekly
Monthy
Form of Results:
Remote Newspaper
Email Newspaper
OK
Reset
Cancel
Help

FIG.2D

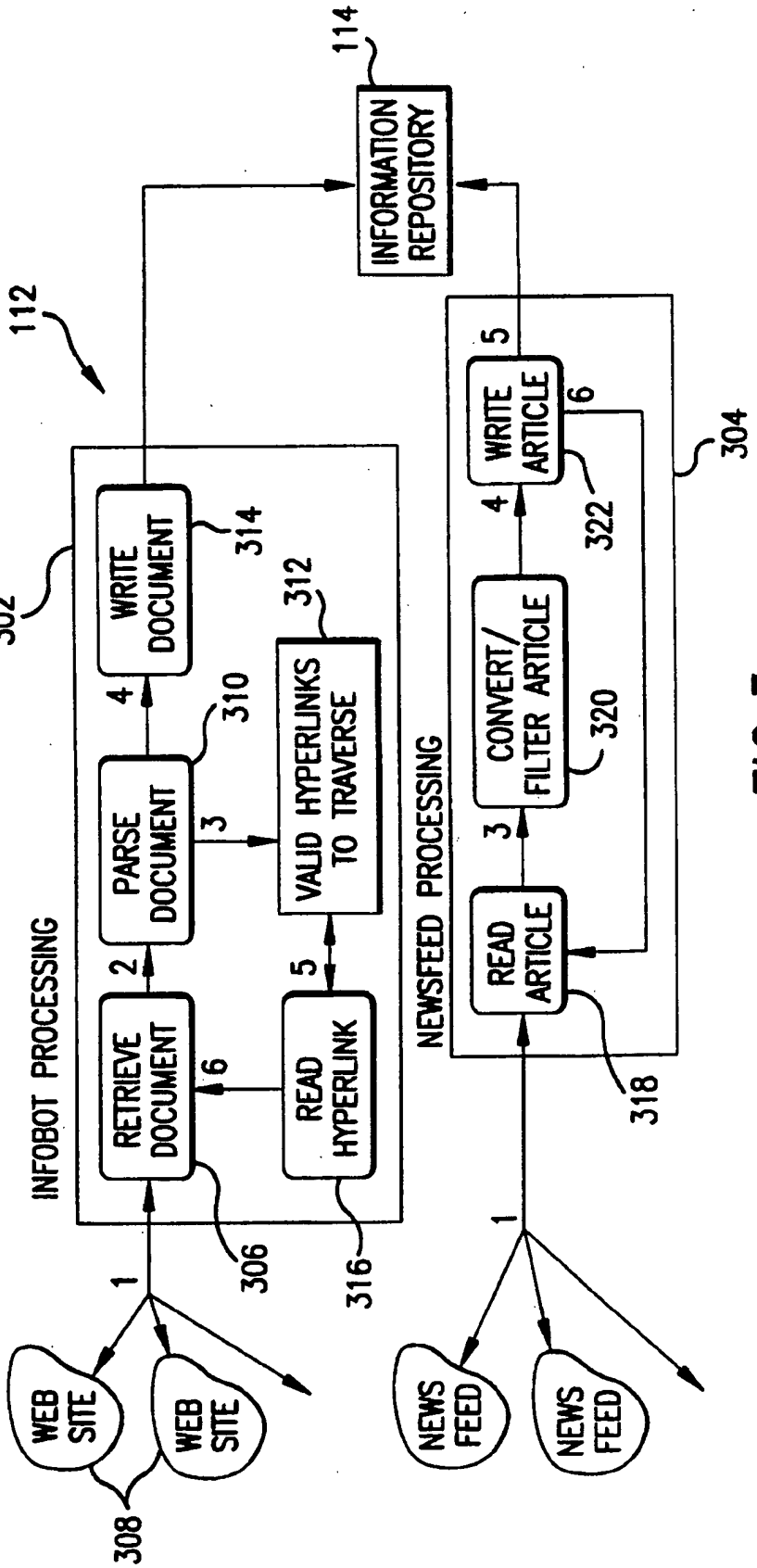

114
INFORMATION REPOSITORY
112
302
WRITE DOCUMENT
314
4
PARSE DOCUMENT
310
3
VALID HYPERLINKS TO TRAVERSE
312
5
2
RETRIEVE DOCUMENT
6
READ HYPERLINK
INFOBOT PROCESSING
306
316
1
WEB SITE
WEB SITE
308
NEWSFEED PROCESSING
304
WRITE ARTICLE
5
6
322
4
CONVERT/ FILTER ARTICLE
320
3
READ ARTICLE
318
1
NEWS FEED
NEWS FEED

FIG.3

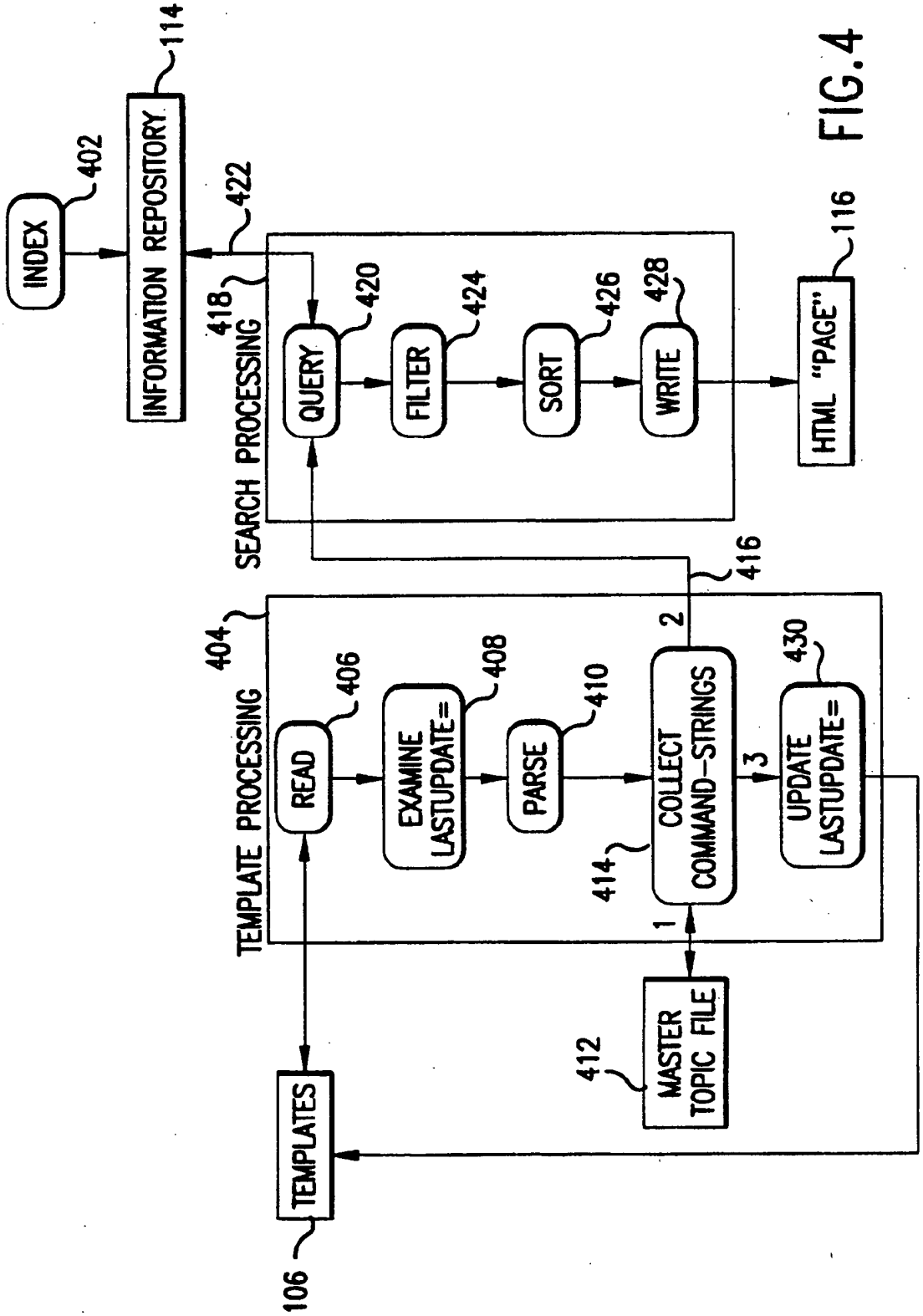

INDEX
402
INFORMATION REPOSITORY
114
422
SEARCH PROCESSING
418
QUERY
420
FILTER
424
SORT
426
WRITE
428
HTML "PAGE"
116
416
TEMPLATE PROCESSING
404
READ
406
EXAMINE LASTUPDATE=
408
PARSE
410
COLLECT COMMAND-STRINGS
414
2
3
1
UPDATE LASTUPDATE=
430
MASTER TOPIC FILE
412
TEMPLATES
106

FIG. 4

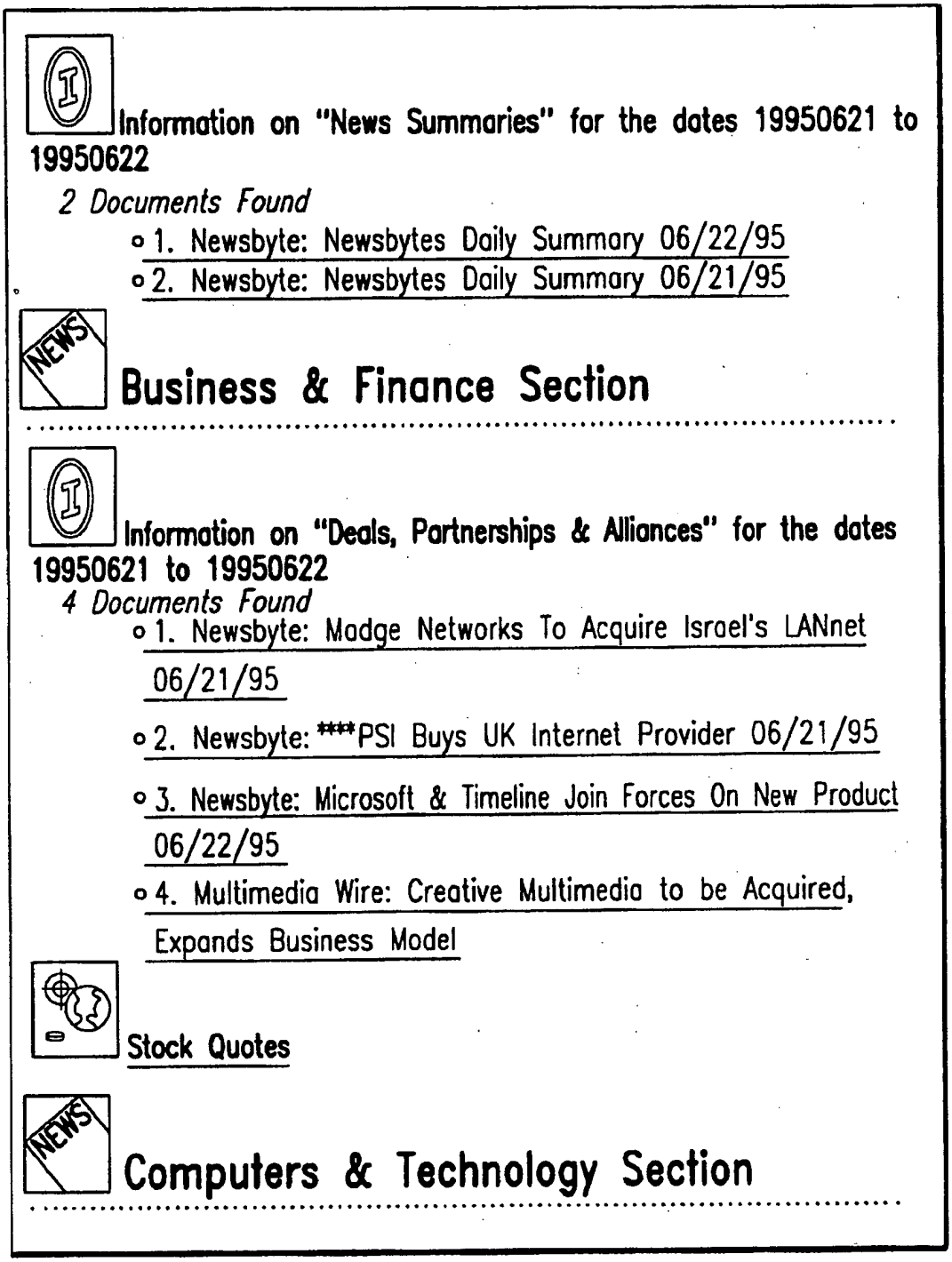

Information on "News Summaries" for the dates 19950621 to 19950622
2 Documents Found
1. Newsbyte: Newsbytes Daily Summary 06/22/95
2. Newsbyte: Newsbytes Daily Summary 06/21/95
Business & Finance Section
Information on "Deals, Partnerships & Alliances" for the dates 19950621 to 19950622
4 Documents Found
1. Newsbyte: Madge Networks To Acquire Israel's LANnet 06/21/95
2. Newsbyte: ****PSI Buys UK Internet Provider 06/21/95
3. Newsbyte: Microsoft & Timeline Join Forces On New Product 06/22/95
4. Multimedia Wire: Creative Multimedia to be Acquired, Expands Business Model
Stock Quotes
Computers & Technology Section

FIG.5

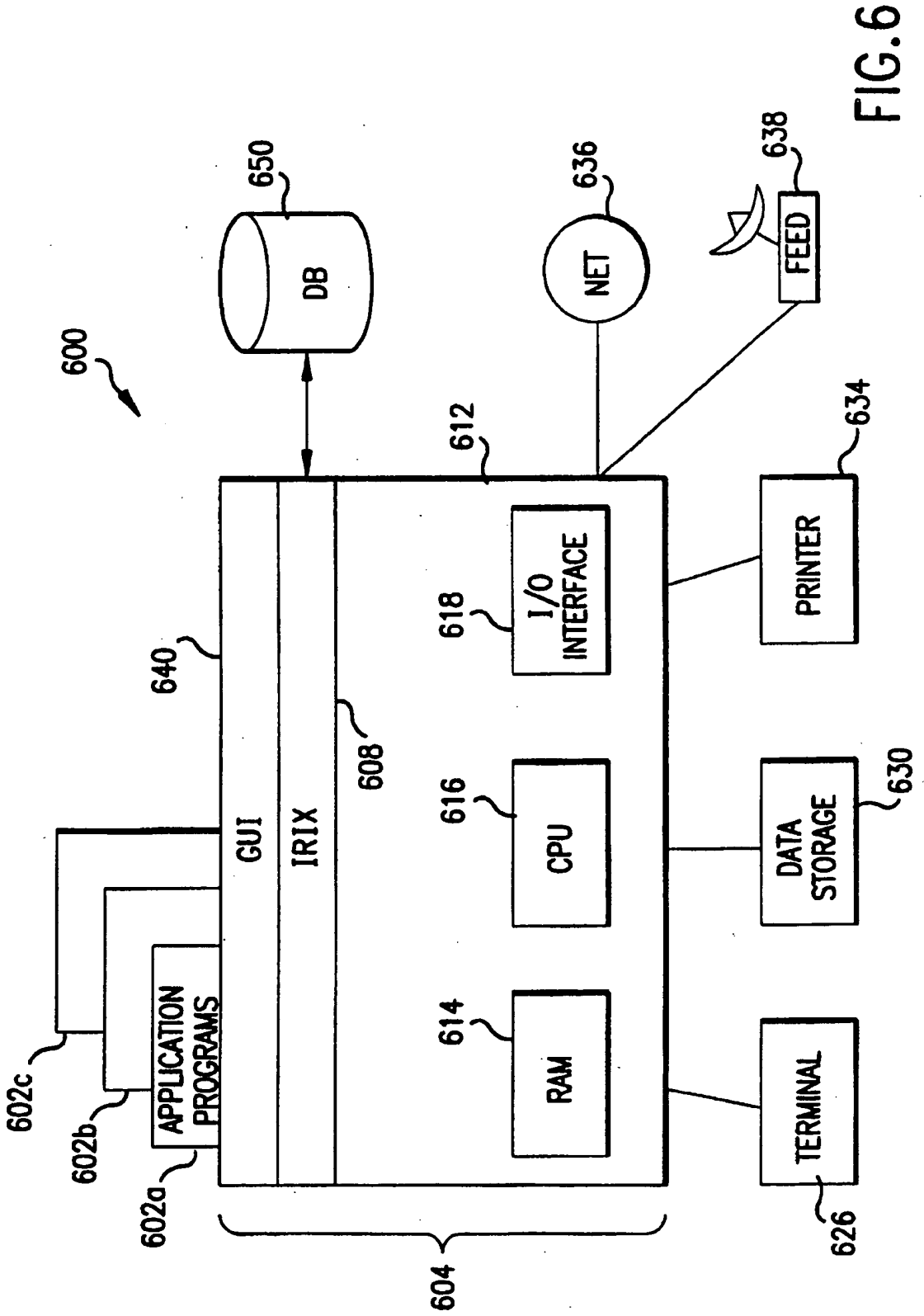

600
650
DB
636
NET
638
FEED
612
634
PRINTER
618
I/O INTERFACE
640
608
616
CPU
630
DATA STORAGE
GUI
IRIX
614
RAM
602c
602b
APPLICATION PROGRAMS
602a
626
TERMINAL
604

FIG.6

SYSTEM AND METHOD FOR A COMPUTER-BASED DYNAMIC INFORMATION CLIPPING SERVICE

BACKGROUND OF THE INVENTION

1. Field of the Invention

The field of the invention relates generally to accessing information on a network, and more particularly, to a system and method providing a dynamic information clipping service.

2. Related Art

Computer networks and on-line services, such as the Internet, have become a common source of news and information for computer end-users. The Internet's size and (lack of) organization, however, make repeated accesses to, and sorting of, data on a periodic basis very time consuming.

Programs have been developed that perform automatic searches for end-users to retrieve information based on specific search queries. These programs merely return search results as files for consumption (e.g., reading) by the end-user. The data returned by these programs is in its original format, which varies greatly from item to item (and from service-to-service). The various item/document formats complicates reading them.

Commercial databases, such as Lexis/Nexis™, Orbit™, Dialog™ and the like, are separate from the Internet and provide some form of item/document formatting when search results are displayed to the end-user. These services are very expensive. To reduce costs for repeat searches, some commercial databases provide other search services that automatically perform update searches periodically. In this case, the search query is saved by the service provider's system.

The same search is repeated at time intervals specified by the end-user, and the results are forwarded to the end-user automatically. However, if terms used to formulate a search query are not accurate, or the subject matter of the topic has developed new terminology or is otherwise divergent, the search query becomes stale. Thus, the results of the subsequent repeat searches can become inaccurate; decreasing both the precision of the search, and the recall of the information by utilizing such queries.

What is needed for Internet end-users is an accurate technique/service for accessing information on "the net" with a minimum level of user specificity and involvement, while being cost and time efficient.

SUMMARY OF THE INVENTION

The present invention is directed to a system and computer-based method providing a dynamic information clipping service. An end-user creates a template of topics of interest via a graphical user interface. The template is transmitted to a central site for processing. At the central site, data is collected that relates to a particular base of knowledge. The data is then parsed, indexed and stored in an information repository.

Processing of the template comprises parsing it, collecting command-strings relating to the parsed template, and querying the information repository using the collected command-strings to generate query results. The query results are then sorted. A HyperText Mark-up Language (HTML) page is created using the sorted query results. The HTML "page" is delivered or otherwise made available on a periodic basis to the end-user for viewing, wherein the HTML page represents a custom network-based newspaper.

A preferred embodiment of the invention is a system and method that provides a dynamic information clipping service for the Internet.

In a preferred embodiment of the invention, collecting data includes using an infobot responsive to Uniform Resource Locators (URLs) to traverse hyperlinks associated with a particular base of knowledge.

In a further embodiment, the collecting includes the creation, and maintenance of a master topics file. This includes creating and assigning keys to each entry in the template, comparing the keys to the master topics file. If a match is found, that command-string is retrieved from the master topics file used for querying (i.e., searching) of the information repository and then adding the results of the query to the end-user's page that corresponds to the template being processed.

Modification to the master topics file is done in a manner that is transparent to the end-user, so as to provide more accurate and current information to the end-user without requiring the end-user to modify the template.

BRIEF DESCRIPTION OF THE FIGURES

The present invention will be described with reference to the accompanying drawings, wherein:

FIG. 1 shows a high-level view of the process according to the present invention.

FIGS. 2A, 2B, 2C and 2D show various features of an exemplary NewsEditor application window **200**, according to a preferred embodiment of the present invention.

FIG. 3 shows more detail of the information collection process **112** of FIG. 1, according to a preferred embodiment of the present invention.

FIG. 4 shows more detail of information processing phase **112** of FIG. 1, according to a preferred embodiment of the present invention.

FIG. 5 shows a resultant "page," according to a preferred embodiment of the present invention.

FIG. 6 shows a general hardware environment in which a preferred embodiment of the present invention can operate.

The preferred embodiment of the invention is described below with reference to these figures where like reference numbers indicate identical or functionally similar elements. Also in the figures, the left most digit of each reference number corresponds to the figure in which the reference number is first used.

DETAILED DESCRIPTION OF THE PREFERRED EMBODIMENTS

The present invention provides a user-friendly method for constructing a "template" that dictates the type of information an end-user is interested in. The invention includes a series of back-end processes that collect, categorize, filter, search, retrieve, and assemble the desired information into a HyperText Mark-up Language (HTML) "page". The invention also includes a method for viewing such a "page" through a Web-browser, such as Netscape Communications Corporation's Netscape™ browser.

As used below, WWW stands for "World Wide Web." The WWW project, started by CERN (the European Laboratory for Particle Physics), seeks to build a distributed hypermedia system. The WWW, also referred to as the "Web," can be termed a client-server based, information presentation system in which everything is a (possibly) hypertext document that may be searchable.

URL is a draft standard for specifying an object on the Internet, such as a file or newsgroup. The following are URL formats (file: and ftp: URLs are synonymous):

file ://wuarchive.wustl.edu/mirrors/msdos/graphics/gifkit.zip

ftp://wuarchive.wustl.edu/mirrors http://www.w3.org:80/default.html

news:alt.hypertext

telnet://dra.com

The first part of the URL, before the colon, specifies the access method. The part of the URL after the colon is interpreted specific to the access method. In general, two slashes after the colon indicate a machine name (machine: port is also valid). In general, documents on the WWW are written in HTML.

An environment created according to the present invention can best be described as an Internet-based, dynamic, "information clipping service." FIG. **1** shows a high-level view of the process according to the present invention. (In the operational diagrams of the invention, rectangles with square corners represent dam stores, such a templates or pages. Rectangles with smoothed corners represent processes, such as template submittal or viewing processes.)

Template Submittal

The submittal process is very straight-forward: a graphical user interface (GUI; not shown) is run by an end-user **102** to choose topics of interest, broken down by section. Specific Web sites (addressed by their URLs/Uniform Resource Locators) can be used within the resultant template as well. The template, upon the end-user saving it at the end-user's site (see template submittal process **104**), is then transmitted to a central site for processing. At the central site, the transmitted template is read and stored as a file in a templates store **106**. In connection with the present invention the GUI is called a "NewsEditor." An exemplary NewsEditor GUI is shown in FIG. **2**. The features of the NewsEditor GUI/template will be described in detail below.

Information Collection

Turning again to FIG. **1**, information collection **108** and information processing **110** are important aspects of the system. Without the appropriate information, the resultant "page" will hardly be worth reading. The information collection process collects data in the form of "sources" and "feeds" **112**. This aspect of the system currently comprises of two components: a "Web-crawler," also called an "infobot," combs (i.e., searches) selected areas of the Web and catalogs documents for eventual indexing; and a capability that allows for certain "newsfeeds" to enter the system and therefore possibly become a part of an end-user's page.

Exemplary newsfeeds include Associated Press Inc. (API) wire services and MultimediaWire™, which is transmitted to the control site via Internet email from Bethesda, Md. Both the infobot and the newsfeeds will be described in detail below.

Collected information is stored in an information repository **114**. The information processing **108** correlates end-user templates in template store **106** with the information in the information repository **114** to create the end-user's page **116**. The end-user implements a view process **118** to read the page.

The present invention permits templates of many end-users to be serviced by one or more central sites. The central site(s) process templates and collected information at different times, depending on end-user specified variables, and the rate at which information is updated in the information repository **114**. Thus, the template submittal process **104**, information collection process **110**, information processing **108**, and view process **118** can all run independently or in parallel with each other. For example, template submittal can be done at any time, even when information is being processed for the templates previously stored at a central site.

FIG. **2A** shows an exemplary NewsEditor application window **200**. Different sections that are available to the end-user to select are displayed at an options menu button **202** titled "Section." Listed on the options menu button **202** is the currently selected section. In the example, the currently selected section is titled "General News." (Other example sections include the following: Business & Finance; Computers & Technology; Film, Video & Broadcast; Games & Interactive Media; and Advertising.) A portion of the entries available under the section General News are displayed in a large window **204** (called the entries window) below the section heading. The options menu button expands into a list of available sections when clicked-on by the end-user, as shown at **205** of FIG. **2B**. Once the section options menu is expanded, the user may click on another section to view its entries in window **204**.

One or more entries can be selected by the end-user by clicking on the desired item, or by dragging a "rubberband" around a group of items to select them. As shown in FIG. **2C**, once one or more items **206** in the entry window **204** are selected, the end-user simply clicks on the "Add to Newspaper" button **208** to add these items to his custom newspaper template. Other methods of selecting an entry by the end-user will become apparent to those skilled in the GUI art.

Once one or more entries are selected, the NewsEditor application then automatically adds the selected entries to the custom newspaper template and instantaneously displays the template as an outline at a second large window **210** (called the custom newspaper template window) located on the right side of the application window **200**. Alternatively, the end-user drags the selected items **206** and drops them into the custom newspaper template window **210** using the pointer device (e.g., mouse or trackball; not shown), at which point they are added to the custom newspaper template.

The end-user is permitted to name the custom newspaper template via a "Newspaper Title" edit box **212**. Under the "Newspaper" menu item **214** is an "options" button that allow the end-user to set-up where the newspaper will be sent to and to specify the frequency of the updating of the paper (e.g., daily, weekly or monthly). A "NewEditor Options" menu **220** is shown in FIG. **2D**. The operations performed by the "Save," "Edit" and "Delete" buttons, as well as other common functions not shown in the figures will be apparent to a person skilled in the art, and familiar with GUI-based application programs.

The "T" icons represent topics-based entries, and the other icons represent "web-jumpers." Web-jumpers represent specific Internet Web sites (URLs) that can be accessed by adding them to the custom newspaper template. The I-icons perform a structured search using command strings to filter through the information repository, as will be discussed in detail below. Each web-jumper is a hyperlink to a preferred web site that the end-user frequently explores.

FIG. **3** shows more detail of the information collection process **112**. Infobot processing is shown at **302** and newsfeed processing is shown at **304**. The infobot accepts a specific URL (or Web-site identifier) and traverses down through all hyperlinks associated therewith. The initial page, called a "homepage," is retrieved from a web site **308**. The homepage is parsed at **310** by examining each hyperlink in

the homepage to determine if it should be traversed, as shown at step 312. Various checks are performed on each hyperlink to determine if the document pointed to by the hyperlink is "desirable." If so, the hyperlink is written to a stack and the process repeats for all hyperlinks. The resultant document that was retrieved (at a step 306 from the web site 308) is then written to the information repository 114 for later indexing, as shown at a process step 314. The next hyperlink is then popped off the stack and read, as shown at a process step 316. The corresponding document is retrieved (at step 306) and the process of evaluating the hyperlinks repeats, until the stack is cleared. The infobot validates hyperlinks by not traversing any one hyperlink more than once (preventing it from getting caught in a circular "loop").

The newsfeed processing 304 receives/reads (see process step 318) incoming documents, which are in the form of e-mail or direct satellite feeds to the central site, and automatically parses and filters the documents into individual articles, as shown at step 320. Again, these articles are then written to the information repository (see step 322) where they are indexed, making them available for possible inclusion in an end-user's page. The conversion/filter step 320 comprises translating the document/article from its source format into HTML.

Information Processing

Information processing 108 will now be described in more detail with reference to FIG. 4. The first step is to ensure that all documents found in the information repository 114 are properly indexed for retrieval. Indexing of the information in the information repository is shown generally at processing step 402. A third party software package is used to perform this operation (available from Fulcrum Technologies, Inc., Ottawa, Ontario, Canada). This package indexes all relevant words found within each document, and provides a method for reading the indices. After indexing, information is available for possible inclusion in an end-user's page.

The first step in preparing the end-user's "page" is to examine the template file that was submitted. This evaluation is called template processing, and is shown generally at 404. Each template file is read (at a process step 406), and a "lastupdate=" field of the file is queried and compared to the current date, at a process step 408. If the "page" requires updating (based on the end-user's Update Preference; e.g., daily, weekly, or monthly), the file is parsed, at a process step 410.

In parsing a template file, each entry in the file contains a certain "key" value. This key value corresponds to a particular topic-based entry. The key value may correspond to what is contained in a master topic file 412. If the key is found, a processing "command string" is retrieved from master file 412, as shown at a processing step 414. A command string is a collection of query parameters, such as: phrases; information regarding which sources to access for a particular topic; and additionally, the limit on the number of documents that will occur in the resultant set; the sort criteria; and other search related parameters that will be apparent to one skilled in the art of information retrieval. The master topics file is maintained (i.e., created, organized and updated) and resides only on the central machine, allowing easy modification and refinement without end-user intervention. The resultant "command string" can be appended-to by the end-user through the use of the GUI (NewsEditor), although by design, they have no specific knowledge of what is contained in the "command string." Only topic-based entries, not URLs, require interrogation of the master topics file.

Once the command string is retrieved, it is passed (see arrow 416) to the search processing stage, shown generally at 418. The information repository 114 is queried (i.e., searched), as shown at processing step 420. Documents that satisfy the query are returned in a result set, as represented by arrow 422. That result set is filtered (see process step 424) according to what would be deemed of the highest relevance to the query, and the set is then sorted (see process step 426) by date, putting pointers to the most current documents at the top of the result set. Those pointers are manipulated in such a way so that they provide addresses (URLs) to Web-based (Internet) documents. These addresses are also referred to as HTTP (or HyperText-Transfer Protocol) addresses. In specifying an address, URLs are used within the "page" to actually point to the original document that was retrieved during the information collection 112. This permits the system not to have to maintain copies of the documents from the selected Web sites that the infobot processes. Newsfeed documents are stored within the central (Web) site, since these are not Web-based documents currently found on other Web sites, and hence must be maintained locally.

After processing all entries within a template file, the resultant "page" is written out (see process step 428), and is now ready for access (viewing) by the end-user. Finally, the "lastupdate=" field of the template is updated to reflect the current date, as shown at a processing step 430.

Viewing

Viewing the resultant Web-based "page" is done through any Web Browser, such as Netscape Communications Corporation's (Mountain View, Calif. Netscape™ browser. Clicking on a document title (part of the result set for a given topic in the template file) will cause the browser to display the full article. An example (portion) of a resultant "page" is shown in FIG. 5.

Hardware

FIG. 6 illustrates a general hardware environment in which a preferred embodiment of the present invention can operate. The environment 600 of the present invention includes application programs 602*a*, 602*b* and 602*c*. Computer platform 604 includes a hardware unit 612, which includes potentially multiple central processing units (CPUs) 616, a random access memory (RAM) 614, and an input/output interface 618. Computer platform 604 includes an operating system 608. Various peripheral components may be connected to computer platform 604, such as a graphics terminal 626, a data storage device 630, a printing device 634, network 636, and newsfeed 638.

Computer platform 604 is any personal computer, workstation or mainframe computer. In a preferred embodiment, CPU 616 is any processor from the MIPS family of processors including R3000 et. seq. Operating System 608 can be any operating system compatible with computer platform 604. In a preferred embodiment, operation system 608 is the IRIX operating system version 5.3 or greater available from Silicon Graphics. IRIX supports an X System-Windows based graphical user interface (GUI) 640. Operating system 608 must provide a mechanism for multitasking. Operating system 608 is further connected to access a database 650 or other storage media.

The central site and end-user site each comprise hardware such as a environment 600. The end-user site and the central site can be located on the same or separate networks, and thus can be located a great distance apart (i.e., both sites can be independent computer systems having a common network or each can have access to the Internet). In a preferred embodiment, database 650 is configured to store the infor-

mation repository at the central site. The Newseditor/template can comprise an application program at the end-user's site, and the information collection and information processing is implemented as an application program at the central site. Accordingly, only the central site need have the newsfeed via cable, satellite, or the like.

In one embodiment, the present invention is a computer program product (such as a floppy disk, compact disk, etc. also referred to as a computer usable medium) comprising a computer readable media having control logic recorded thereon. The control logic, when loaded into memory **614** and executed by the CPU **616**, enables the CPU **616** to perform the operations described herein. Accordingly, such control logic represents a controller, since it controls the CPU **616** during execution.

Conclusion

While various embodiments of the present invention have been described above, it should be understood that they have been presented by way of example, and not limitation. It will be apparent to persons skilled in the relevant art that various changes in form and detail can be made therein without departing from the spirit and scope of the invention. Thus the present invention should not be limited by any of the above-described exemplary embodiments, but should be defined only in accordance with the following claims and their equivalents. All cited patent documents and publications in the above description are incorporated herein by reference.

What is claimed is:

1. A computer-based method providing a dynamic information clipping service, comprising the steps of:

at an end-user site,

(1) permitting an end-user to create a template of topics of interest via a graphical user interface; and

(2) transmitting said template to a central site for processing; at said central site,

(1) collecting information relating to a particular base of knowledge using an infobot responsive to Uniform Resource Locators to traverse hyperlinks associated with said base of knowledge;

(2) parsing and indexing said information;

(3) storing said parsed and indexed information in an information repository;

(4) processing said template, wherein said processing includes

(a) parsing said template,

(b) collecting command-strings relating to said topics of interest found within said parsed template,

(c) querying said information repository using said collected command-strings to generate query results,

(d) sorting said query results, and

(e) creating a Hypertext Mark-up Language (HTML) page using said sorted query results; and

(5) making said page available to the end-user for viewing, wherein said page represents a custom network-based newspaper.

2. The method of claim **1**, wherein said topics of interest relate to information obtained from at least one of web sites and newsfeeds.

3. The method of claim **1**, wherein said step of making comprises the step of delivering, automatically and periodically according to a period set by the end-user, said HTML page to the end-user for viewing.

4. The method of claim **1**, wherein said step of collecting comprises maintaining information master topics file, and said method further comprises the steps of:

assigning keys to each entry in said template;

comparing said keys to said master topics file; and

if a match is found, retrieving one of said command-strings from the master topics file.

5. The method of claim **4**, further comprising a step of modifying said master topics file in a manner transparent to the end-user, so as to provide more accurate and current information the end-user without requiring the end-user to modify said template.

6. A computer program product for use with a dynamic information clipping service operating on a computer system, said computer program product comprising:

a first computer usable medium having computer readable program code means embodied in said medium for causing an application program to run at an end-user site, said computer readable program code means comprising

(1) a computer readable first program code means for causing the computer system to permit an end-user to create a template of topics of interest relating to information obtained from at least one of web sites and newsfeeds via a graphical user interface; and

(2) a computer readable second program code means for causing the computer system to transmit said template to a central site for processing.

7. The computer program product of claim **6**, further comprising:

a second computer usable medium having second computer readable program code means embodied in said medium for causing an second application program to run at a central site, said second computer readable program code means comprising:

(1) a computer readable third program code means for causing the computer system to collect information relating to a particular base of knowledge;

(2) a computer readable fourth program code means for causing the computer system to parse and indexing said collected information;

(3) a computer readable fifth program code means for causing the computer system to store said parsed and indexed information in an information repository;

(4) a computer readable sixth program code means for causing the computer system to process said template, wherein said processing includes

(a) a computer readable seventh program code means for causing the computer system to parse said template,

(b) a computer readable eighth program code means for causing the computer system to collect command-strings relating to said parsed template,

(c) a computer readable ninth program code means for causing the computer system to query said information repository using said collected command-strings to generate query results,

(d) a computer readable tenth program code means for causing the computer system to sort said query results, and

(e) a computer readable eleventh program code means for

causing the computer system to create a page using said sorted query results; and

(5) a computer readable twelfth program code means for causing the computer system to make said page available to the end-user for viewing, wherein said page represents a custom network-based newspaper.

8. A computer program product of claim **7**, wherein said page is produced in Hypertext Mark-up Language (HTML) format.

9. A computer program product for use with a dynamic information clipping: service operating on a computer system, said computer program product comprising:

a first computer usable medium having computer readable program code means embodied in said medium for causing a first application program to run at a central site, said first computer readable program code means comprising

(1) a computer readable first program code means for causing the computer system to collect information relating to a particular base of knowledge;

(2) a computer readable second program code means for causing the computer system to parse and indexing said collected information;

(3) a computer readable third program code means for causing the computer system to store said parsed and indexed information in an information repository;

(4) a computer readable fourth program code means for causing the computer system to process a template of topics of interest relating to information obtained from at least one web sites and newsfeeds, wherein said processing includes

(a) a computer readable fifth program code means for causing the computer system to parse said template,

(b) a computer readable sixth program code means for causing the computer system to collect command-strings relating to said parsed template from a master topics file,

(c) a computer readable seventh program code means for causing the computer system to query said information repository using said collected command-strings to generate query results,

(d) a computer readable eighth program code means for causing the computer system to sort said query results, and

(e) a computer readable ninth program code means for causing the computer system to create a page using said sorted query results; and

(5) a computer readable tenth program code means for causing the computer system to make said page available to the end-user for viewing, wherein said page represents a custom network-based newspaper.

10. The computer program product of claim 9, wherein said page is produced in Hypertext Mark-up Language (HTML) format.

11. A computer system for providing a dynamic information clipping service, comprising:

end-user site means comprising

(1) first means for permitting an end-user to create a template of topics of interest relating to information obtained from at least one of web sites and newsfeeds via a graphical user interface, and

(2) second means for transmitting said template to a central site for processing; and a central site means comprising

(1) third means for collecting information relating to a particular base of knowledge

(2) fourth means for parsing and indexing said collected information

(3) fifth means for storing said parsed and indexed information in an information repository

(4) sixth means for processing said template, wherein said sixth means includes

(a) seventh means for parsing said template,

(b) eighth means for- collecting command-strings relating to said parsed template,

(c) ninth means for querying said information repository using said collected command-strings to generate query results,

(d) tenth means for sorting said query results, and

(e) eleventh means for creating a page using said sorted query results and

(5) twelfth means for making said page available to the end-user for viewing, wherein said page represents a custom network-based newspaper.

12. The computer system of claim 11, wherein said page is in Hypertext Mark-up Language (HTML) format.

13. The computer system of claim 11, wherein said third means comprises using an infobot responsive to Uniform Resource Locators to traverse hyperlinks associated with said base of knowledge.

14. The computer system of claim 11, wherein said end-user site means and said central site means are independent.

* * * * *